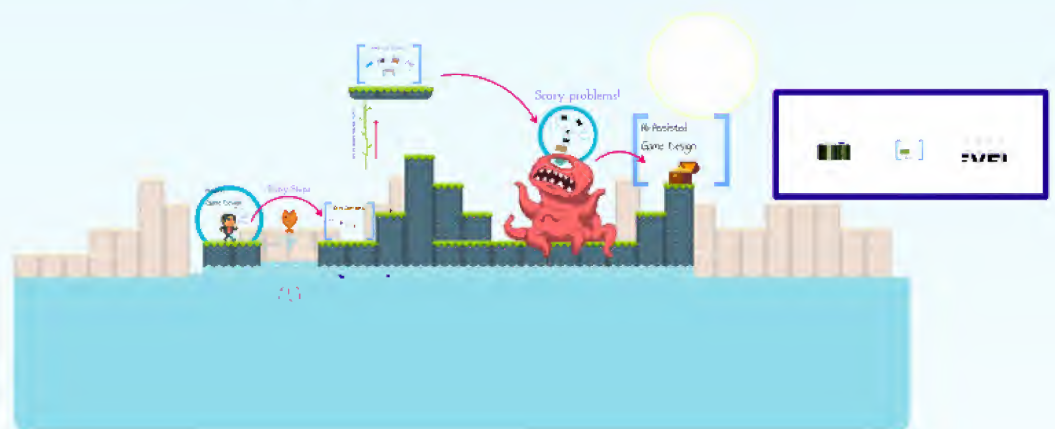




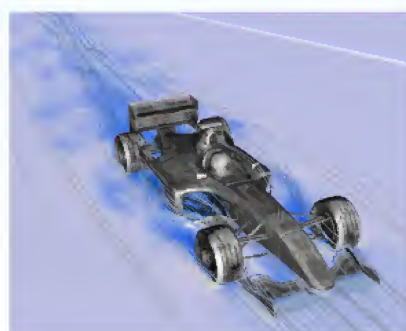
What would it take to make [Shigi](#) a reality?

She's trying to sell you on a way of thinking about concepts

I'm not trying to sell you a book



Virtual Wind Tunnel for Game Design



From the Behavior Up: When the AI Is the Design

Paul Tozour

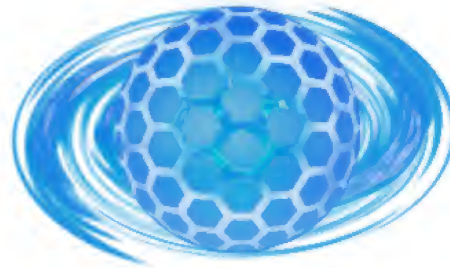
Intelligence Engine Design Systems

Damian Isla

Christian Baekkelund

Moonshot Games





Intelligence Engine

DESIGN SYSTEMS



paultozour.prosite.com
GDC Play kiosk 50

"Evolver" Coevolutionary Optimizer: What, How, and WHY





Shigi



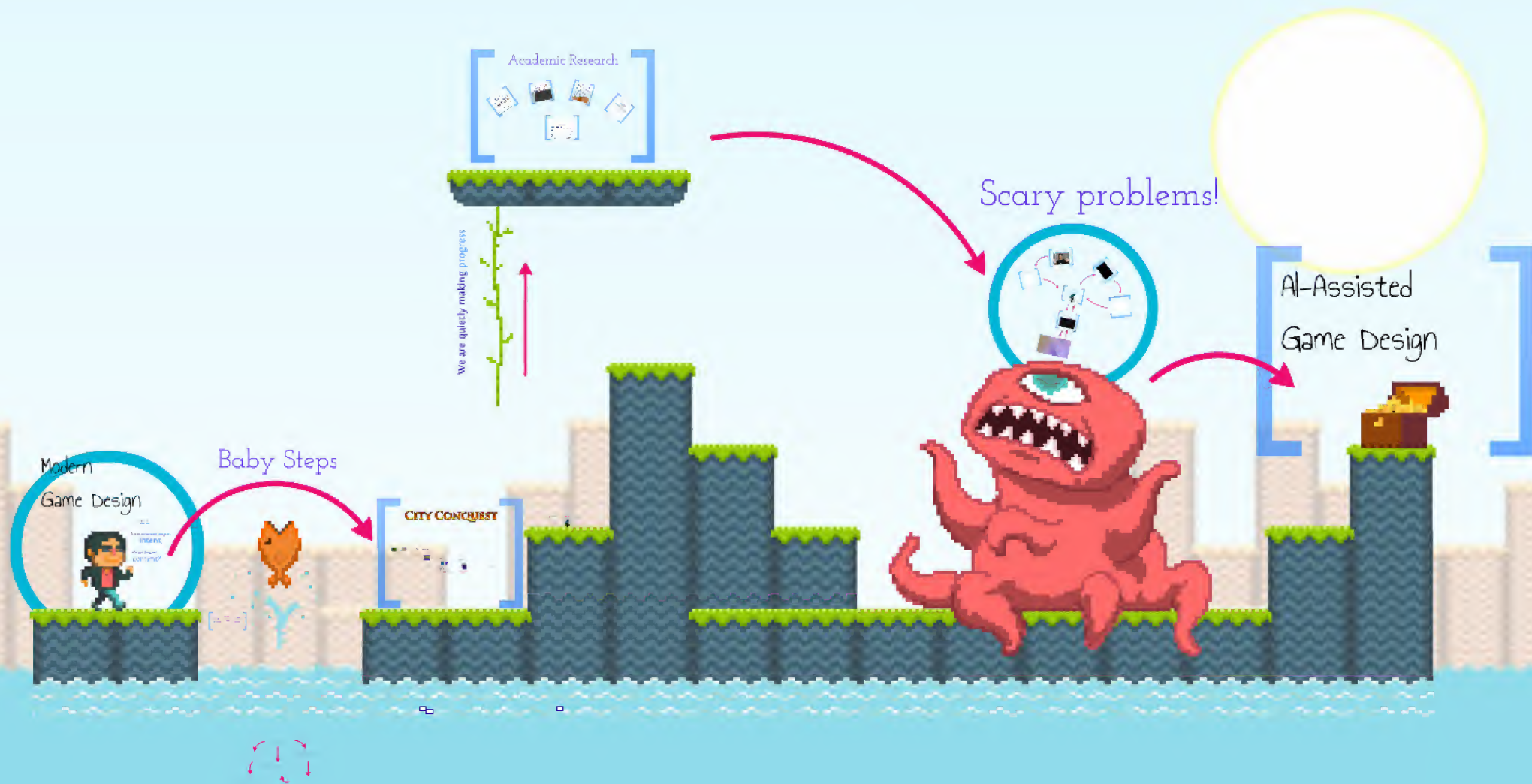
Science
Fiction!

AI-Assisted Game Design



I'm trying to sell you on a way of thinking about design.

I'm not trying to sell you a tool.

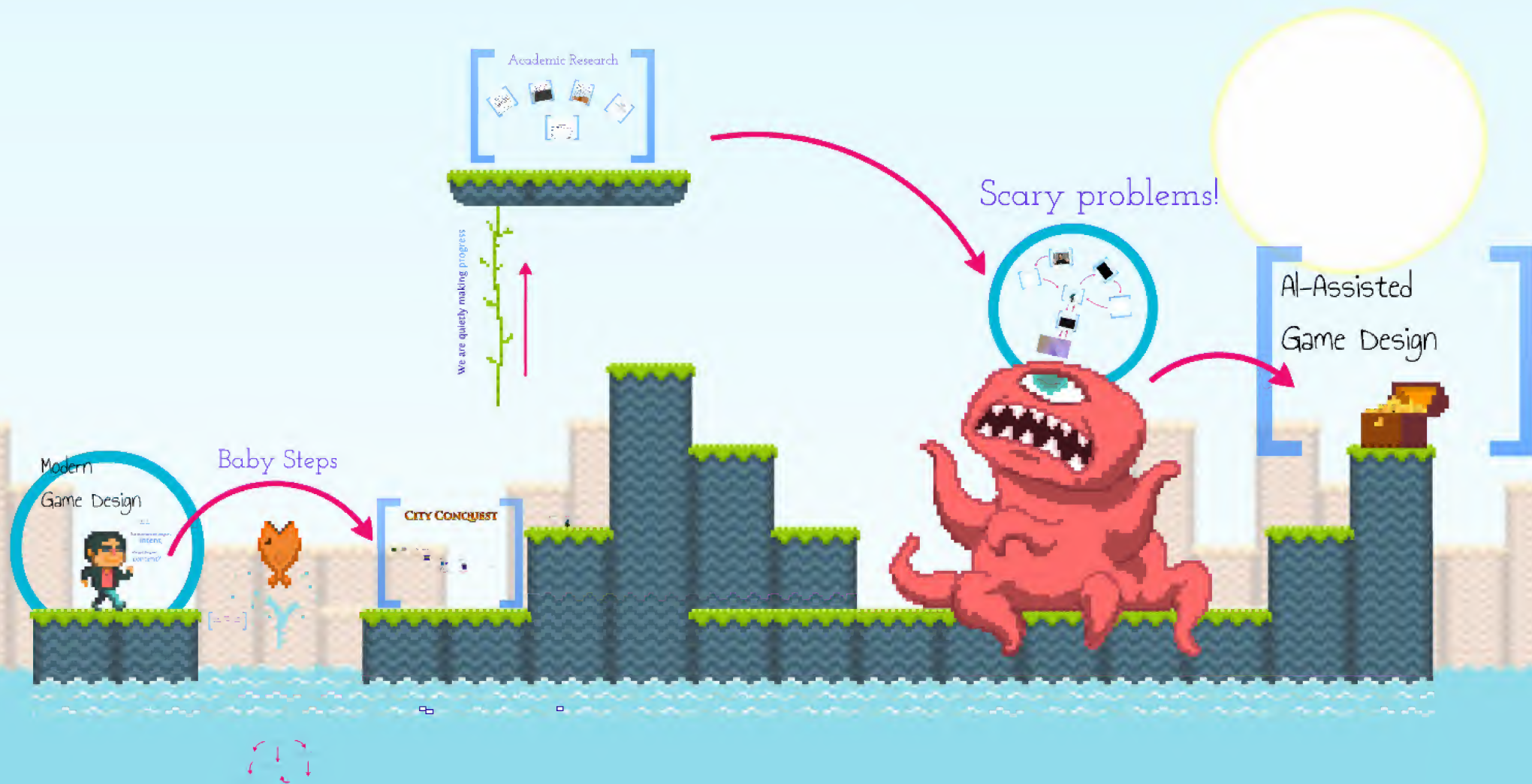


I'm not trying to sell you a tool.

I'm trying to sell you on a
way of thinking about design.

I'm trying to sell you on a way of thinking about design.

I'm not trying to sell you a tool.



Modern

Game Design

What could we do with
"deep tools"

That understand the designer's
intent,

Not just the game's
content?



Lighting
Dialogue Editing
Animation
Scripting
AI
Particle
Level
Sound Des

"We are still alchemists of our trade, mixing two parts impure story with one part polluted gameplay with three parts market voodoo.



As an industry, we need to move beyond the mystical hand waving that defines modern game design."

-- Daniel Cook, Spry Fox

"Our design process is a giant Katmari ball,



picking up every idea it rolls over.”
- Anonymous producer

Modern game design = "iteration"
"Iteration" = trial and error



Exploring the darkness.

No disciplined way to explore the possible design decisions
and their ramifications **without actually implementing them.**

An Edisonian Process



Risky. Expensive. Time-consuming. Error-prone.

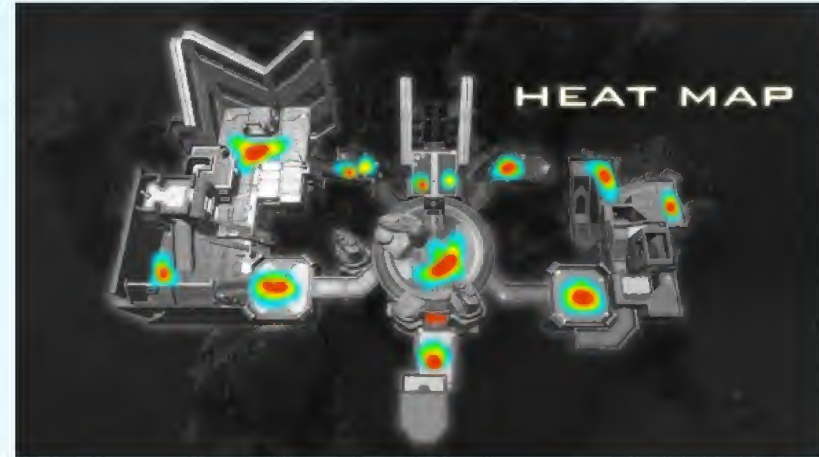
Prototyping is faster & cheaper but STILL very expensive & time-consuming.

We take for granted that this is the ONLY way to design!



HEAT MAP

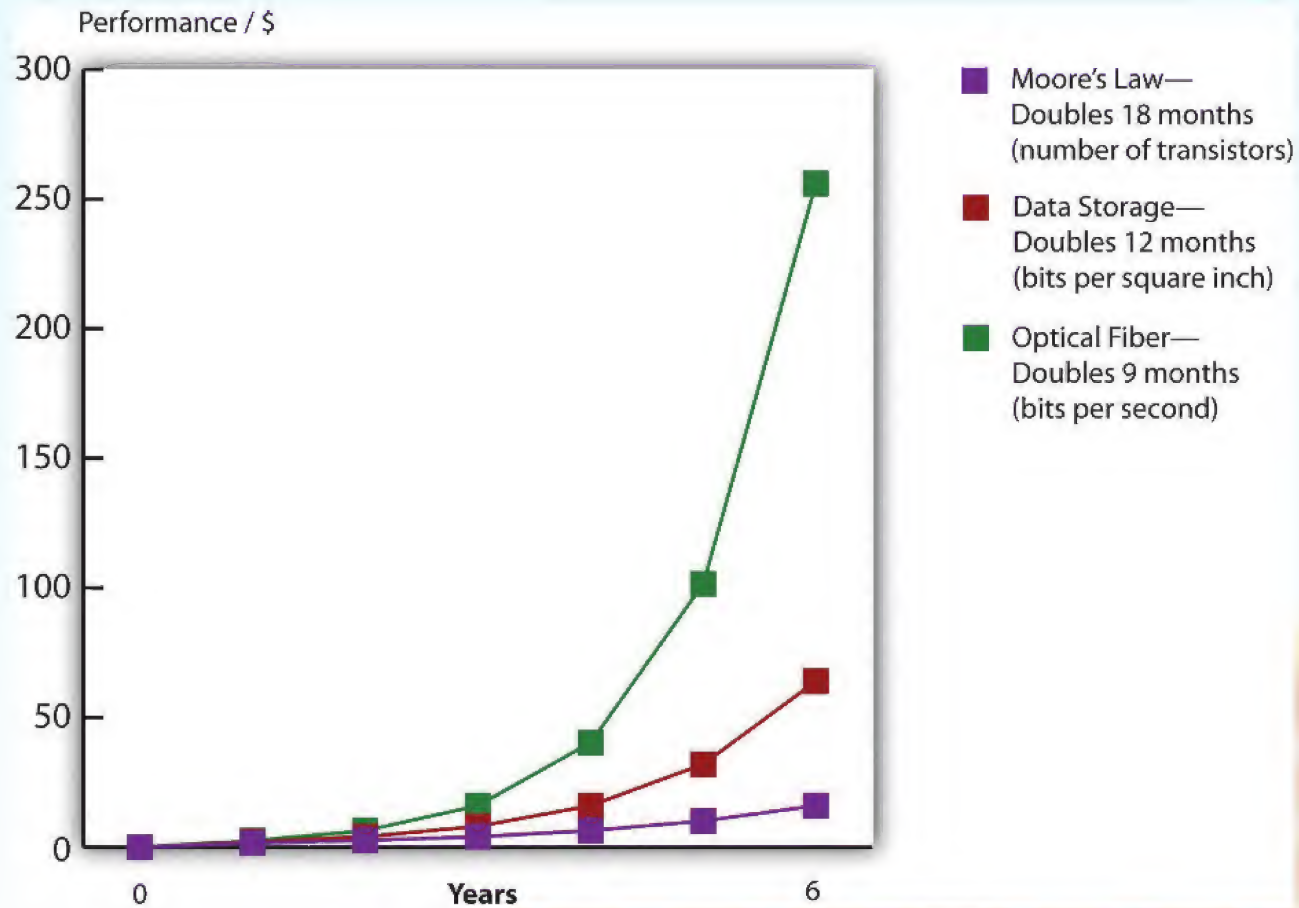




These are not **design** tools.

- Not **interactive**
- Pre-release user testing with focus groups & beta testers is **expensive** -- costs scale linearly
- Post-release user testing is **too late**

Moore's Law



What could we do with
“deep tools”



What could we do with
"deep tools"

That understand the designer's
intent,

Not just the game's
content?



Animation

Lighting

Particle Effects

Scripting

Dialogue Editing

Level Design

AI

Sound Design



I have a discrete description
of the current design



I have a description of your
design goals and constraints

of the



I understand how your game world
represents space and time



I have a c
design go

I have a de
design goal



I understand how the agents can be
created, destroyed, and interact in the
game world



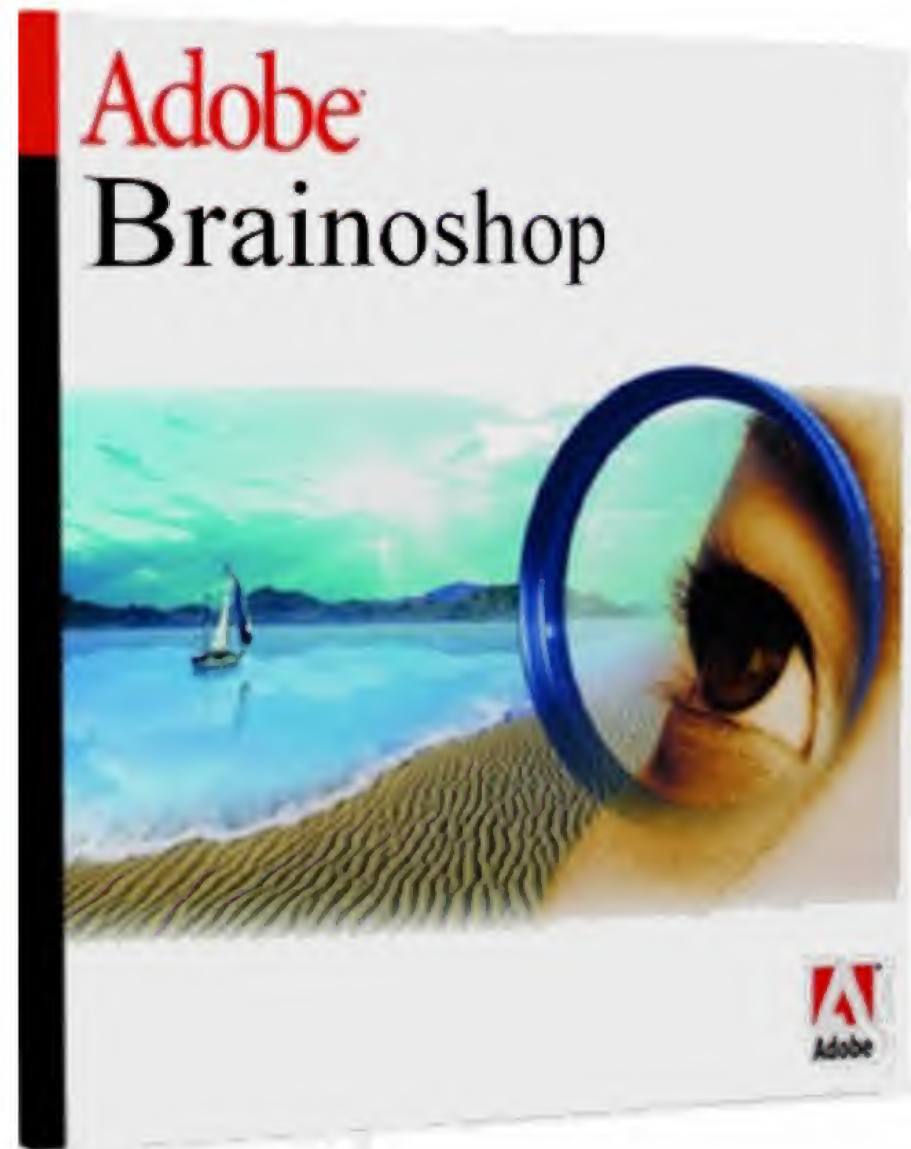
I can EXPERIMENT with the
current design configuration and
test the outcome ...



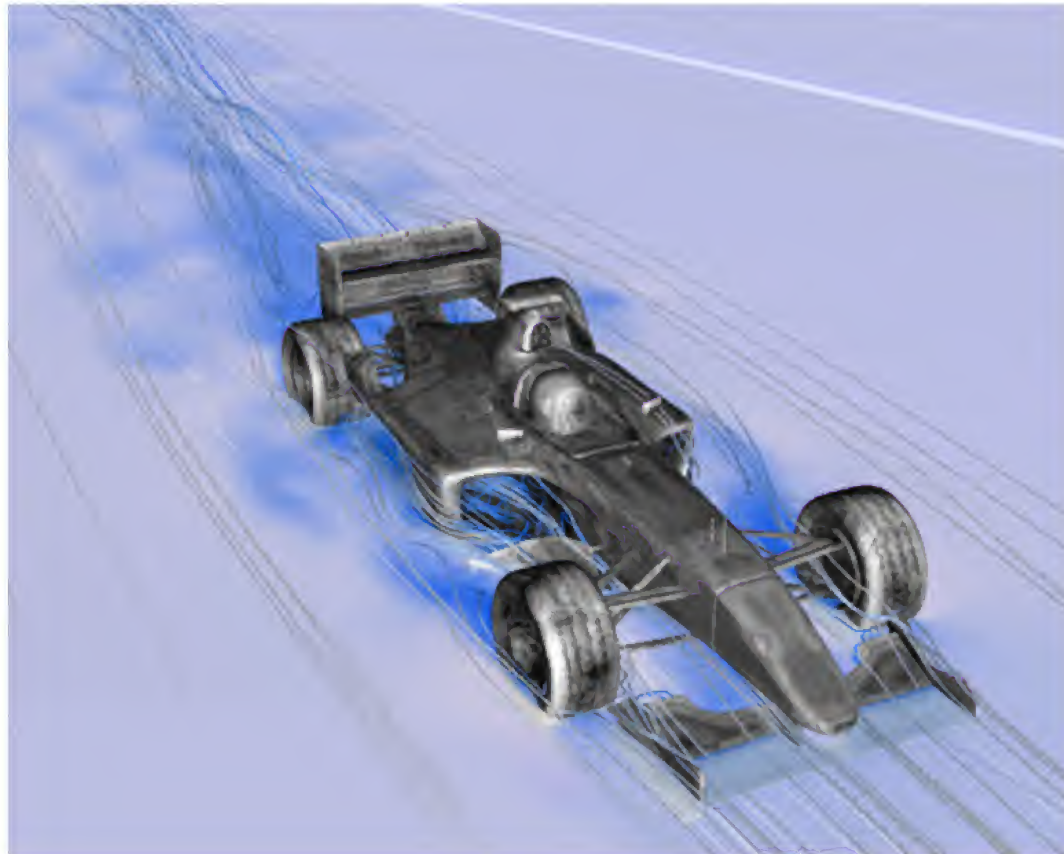


... against your design goals
and constraints

The "Photoshop of AI" Debate



The "Virtual Wind Tunnel for Game Design" Debate



Design is the
process of

Artificial intelligence
is the study &
implementation of

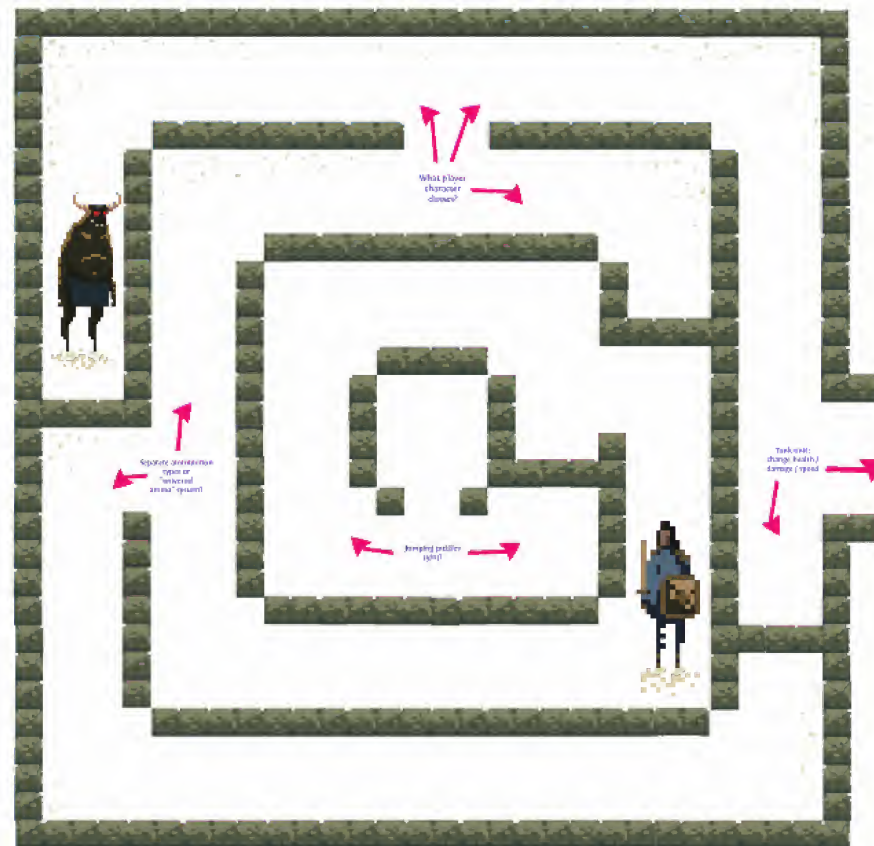
Search.

for optimal design
configurations in the
space of possible
design decisions

to make decisions
more quickly and
more optimally than
humans can do alone.

Design is search.



The process of design is a sequence of decisions through the search space of design possibilities.







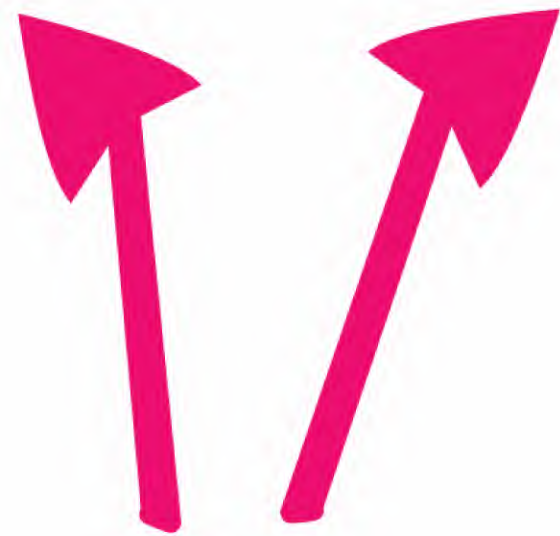
Jumping puzzles
(y/n)?





Separate ammunition
types or
“universal
ammo” system?





What player
character
classes?

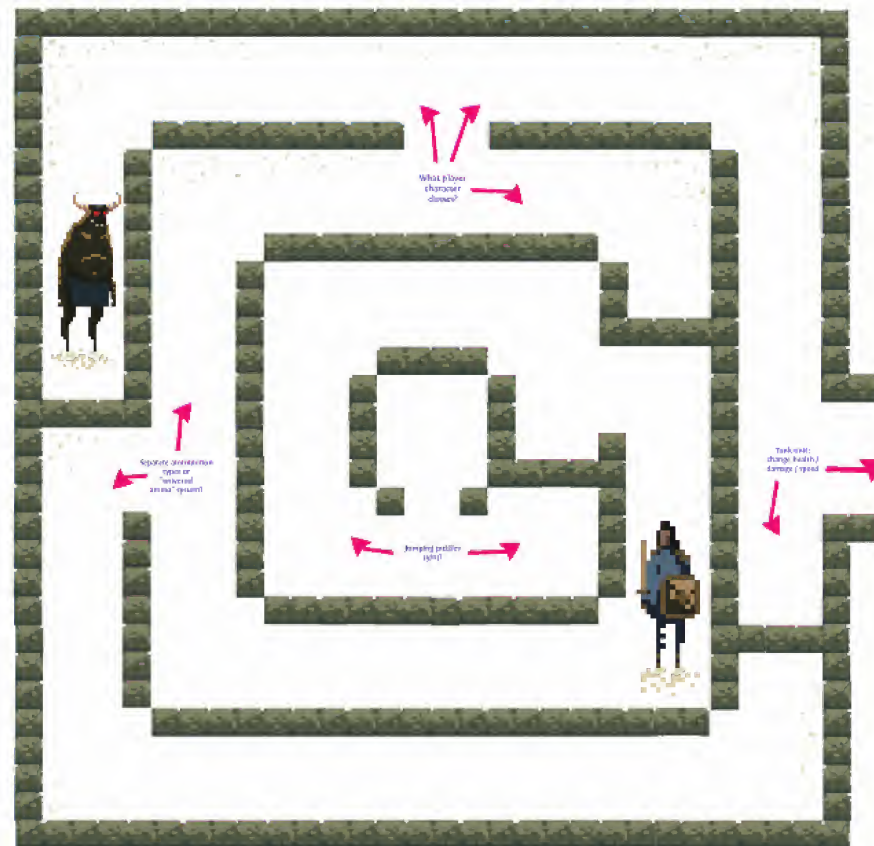


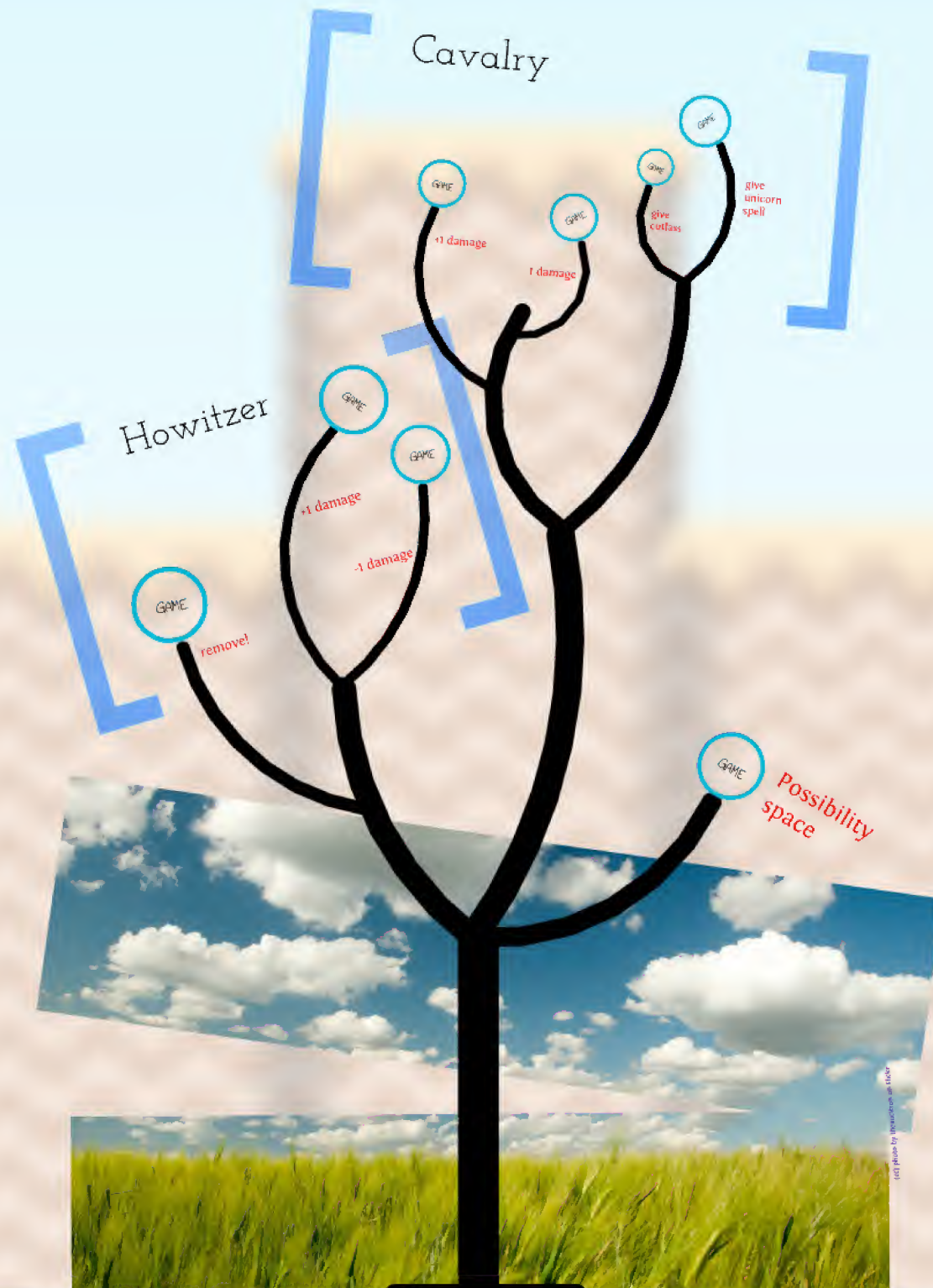
Tank unit:
change health /
damage / speed



Design is search.

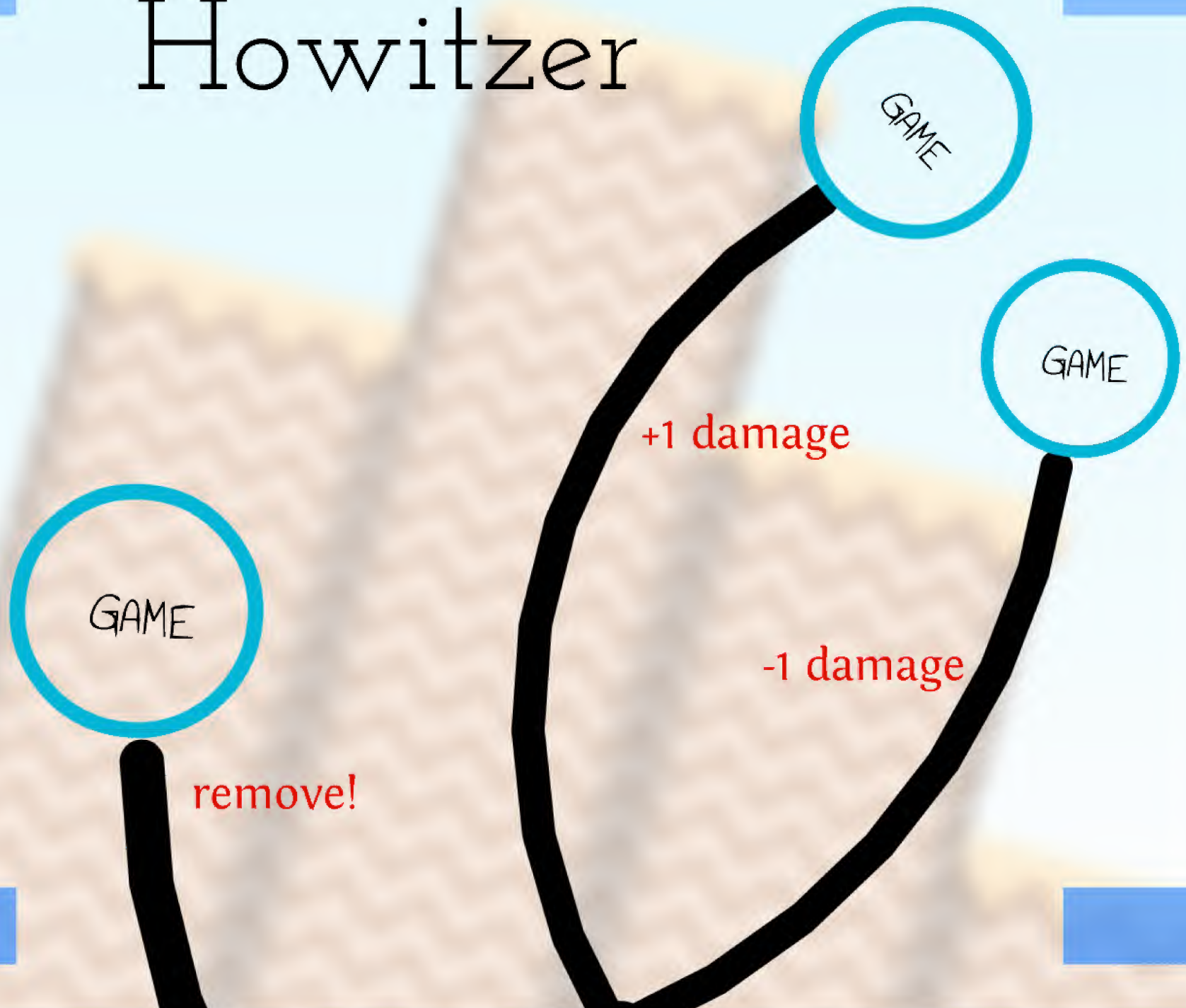
The process of design is a sequence of decisions through the search space of design possibilities.



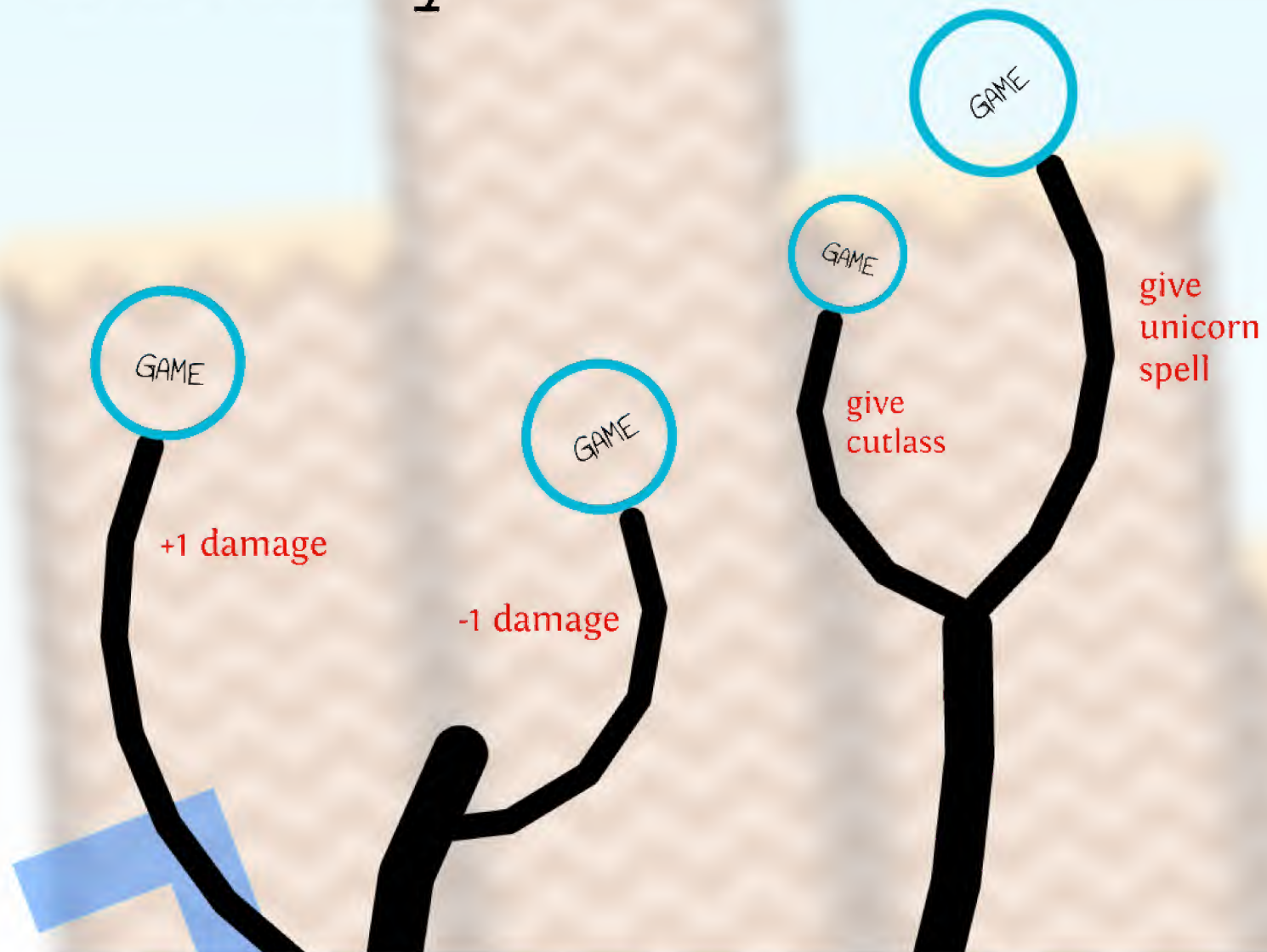


not photo by iStockphoto.com

Howitzer



Cavalry

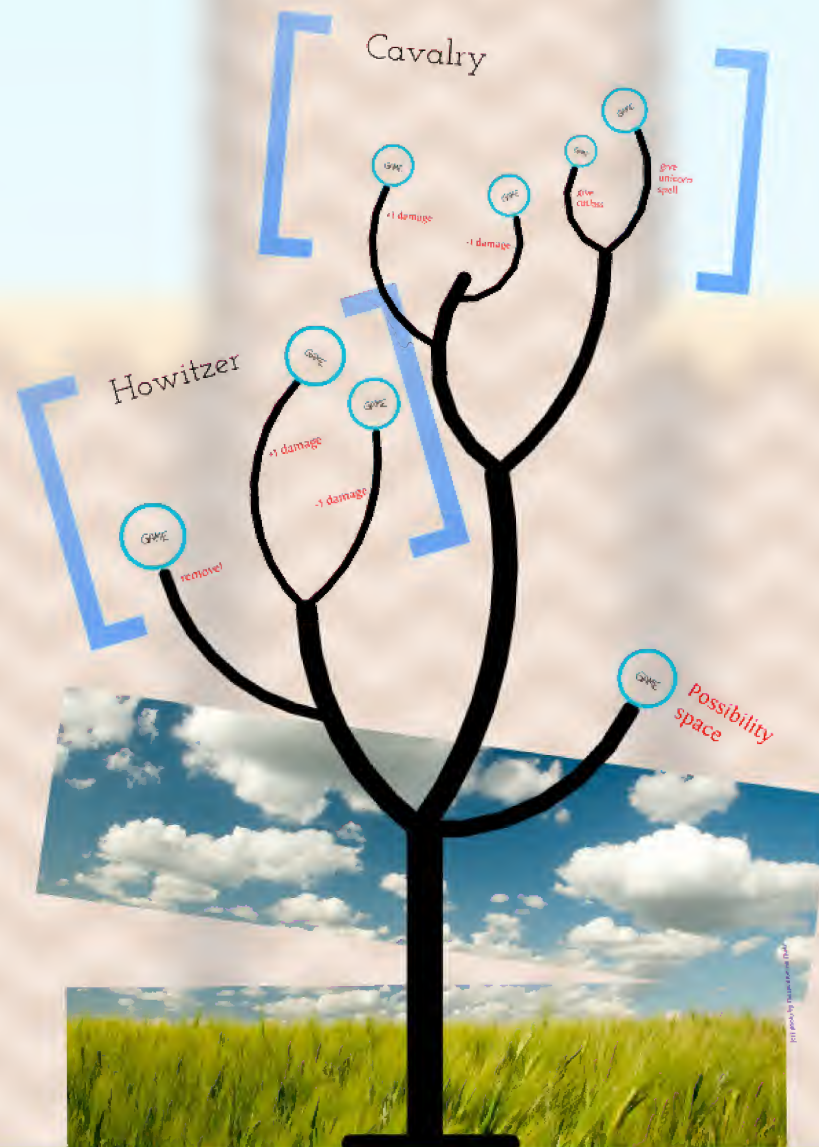




GAME

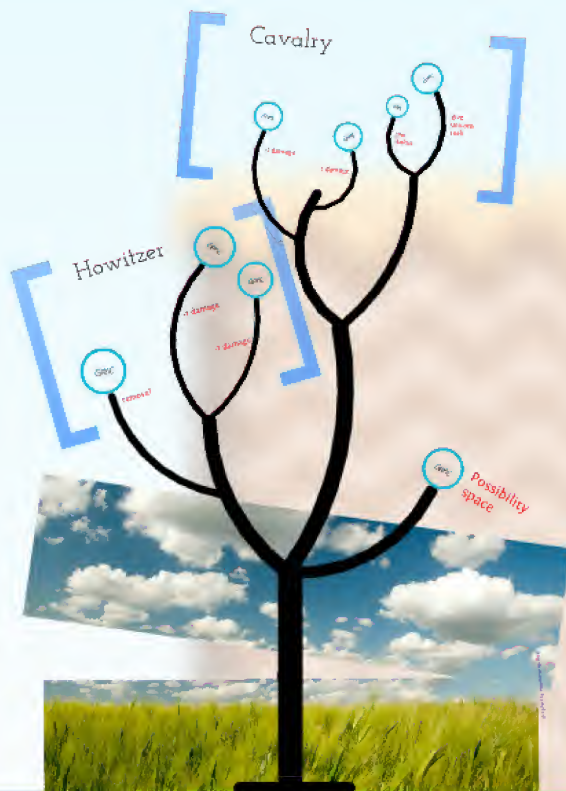
Possibility
space

Design Space



- In theory, automated design could allow us to explore a much larger design space and how it affects game play.
- Challenge: how to quantify and compare different design options.
- Challenge: how to quantify and compare different game components.

Design Space



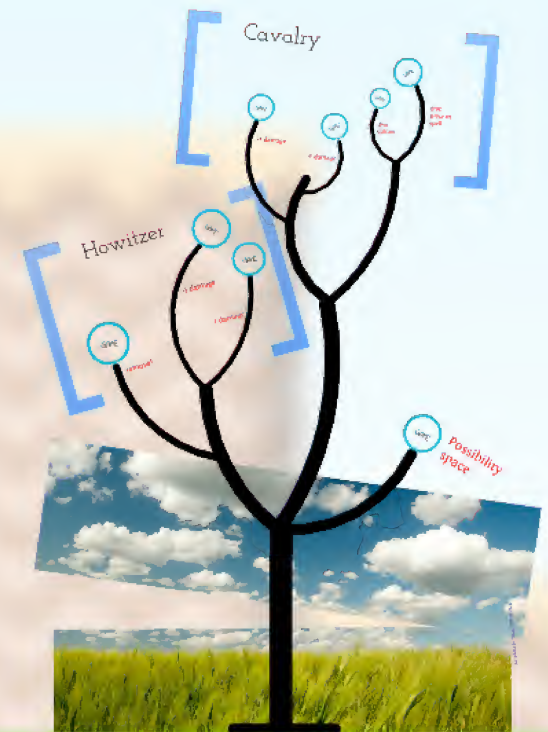
- In theory, you could search this tree to automatically (re)design a game if you could accurately determine **how to branch** and **how to evaluate each branch**.
- Challenge #1: how to **enumerate** the possible design changes
- Challenge #2: how to **characterize** and **quantify** the possibility space of the resulting game configuration

Maybe Not Quite as Hard
as it Looks ...

No need to fully explore each possibility space!
We only care about how it is different from its
parent -- how (or whether) it alters the gameplay.

Don't necessarily need to narrow down to ONE
optimal configuration Search itself can be useful
if it **reduces the scope of the problem** to
something designers can solve manually.

Design Space





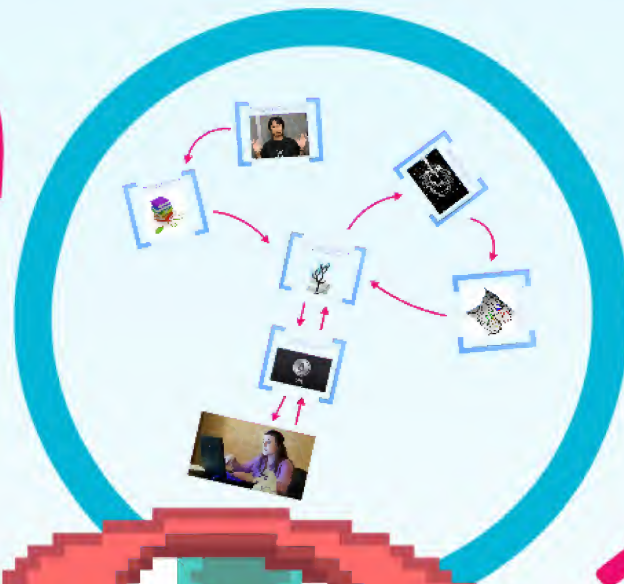
Shigi

What would it take to make **Shigi** a reality?

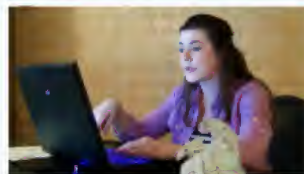
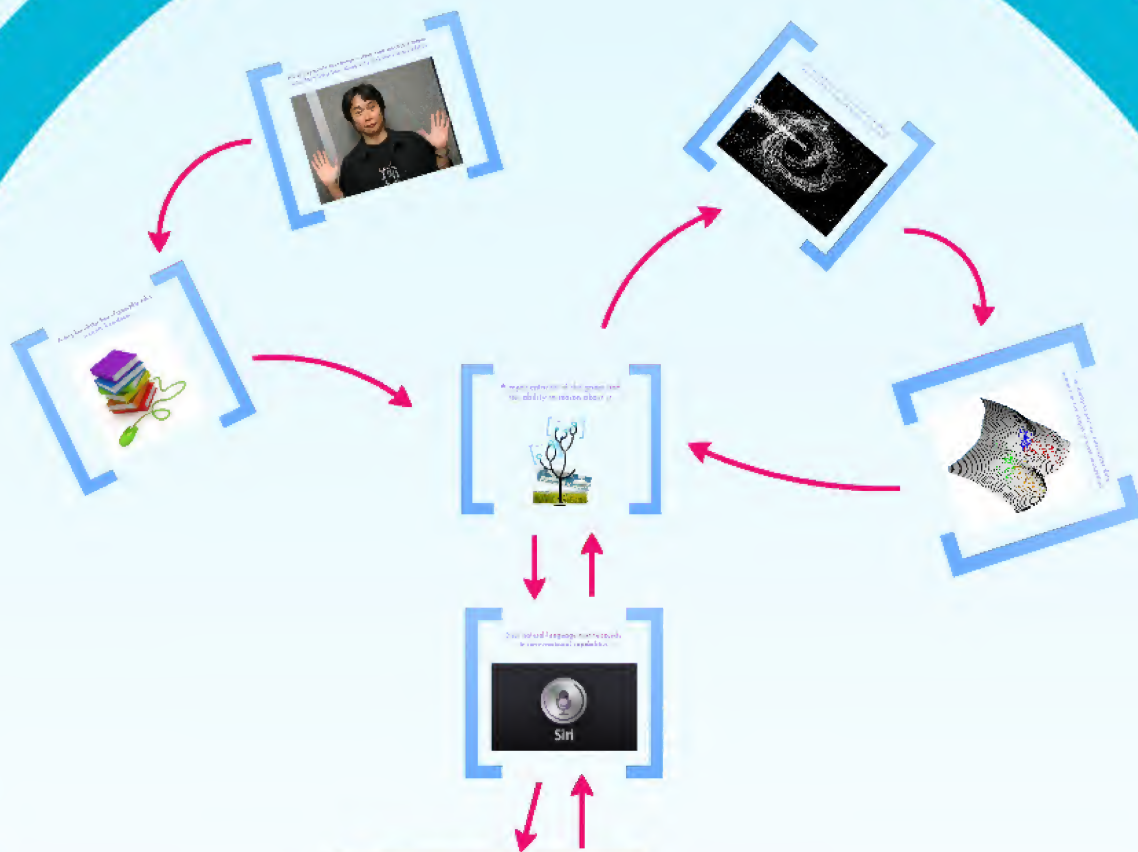
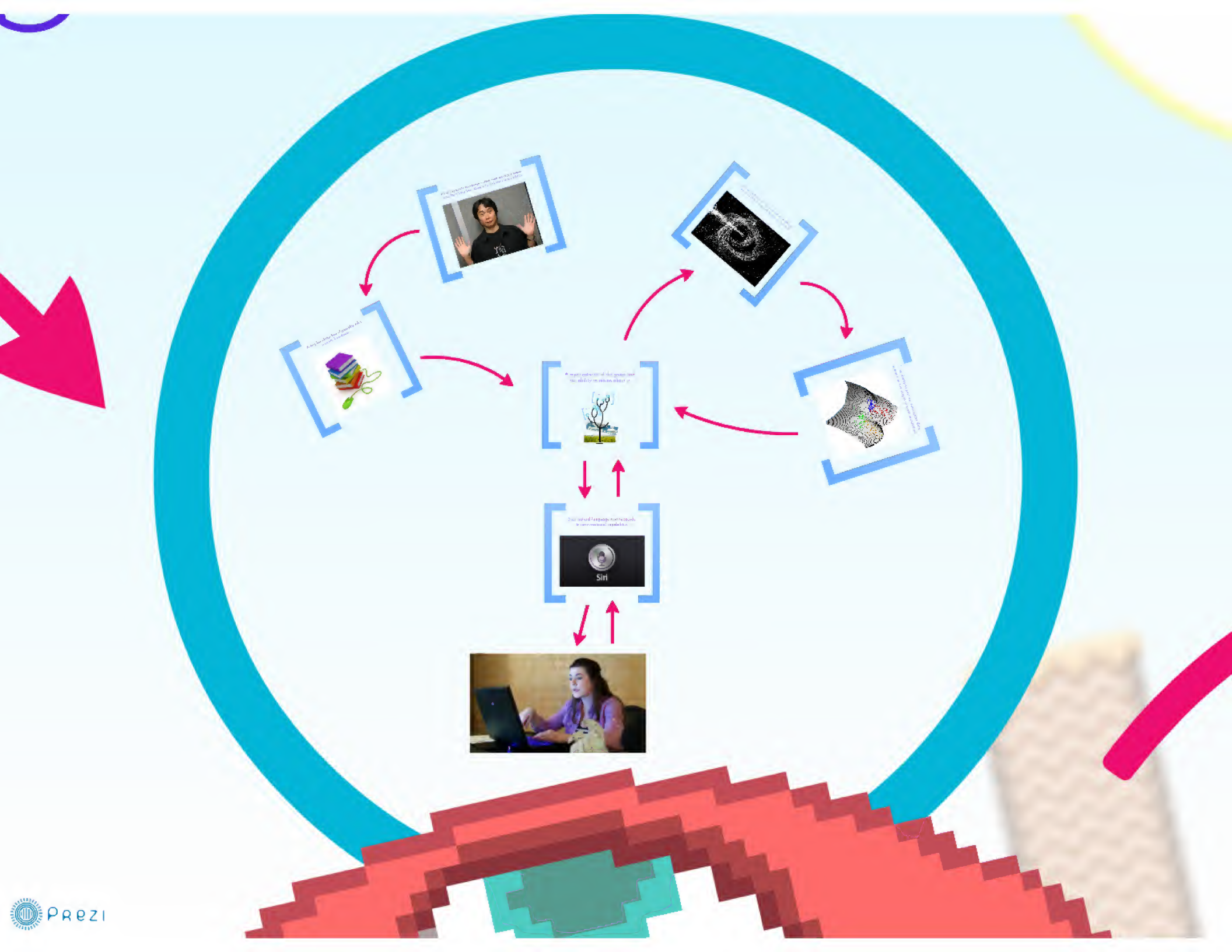
AI-Assisted Game Design

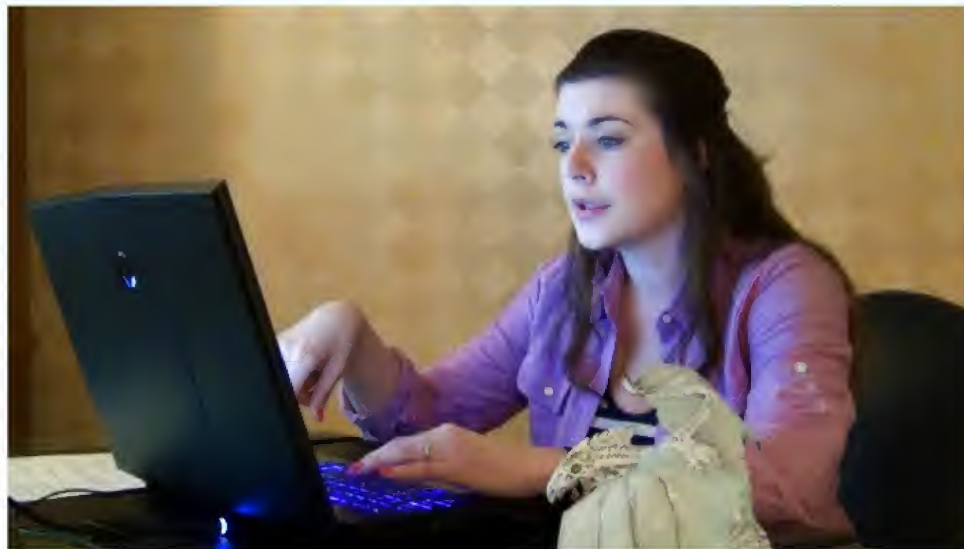


Scary problems!



AI-Assisted
Game Design

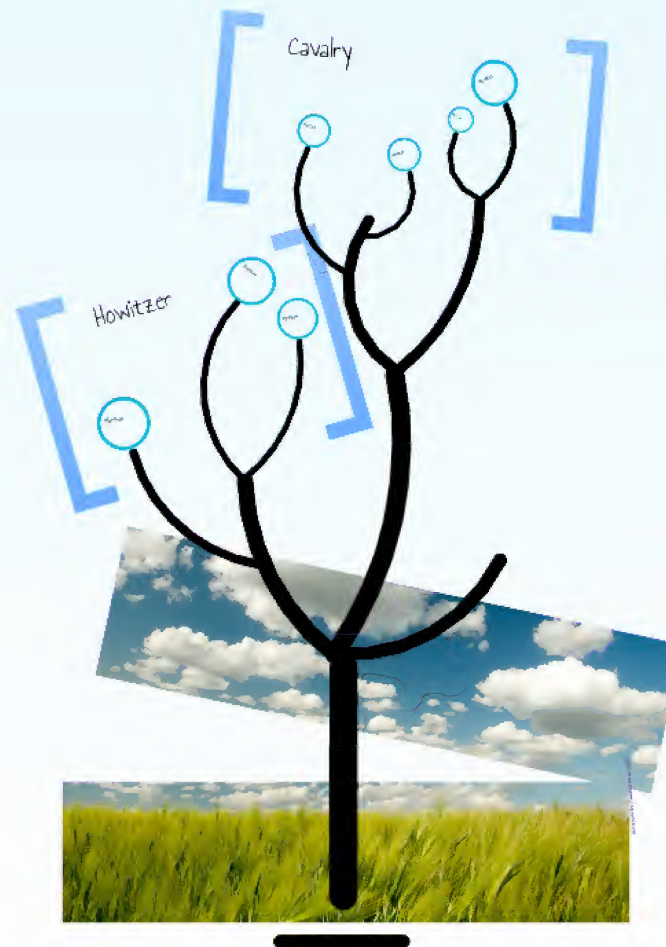




Siri's natural-language, text-to-speech
& conversational capabilities



A representation of the game and
the ability to reason about it



A deep knowledge base of gameplay rules,
concepts, & mechanics



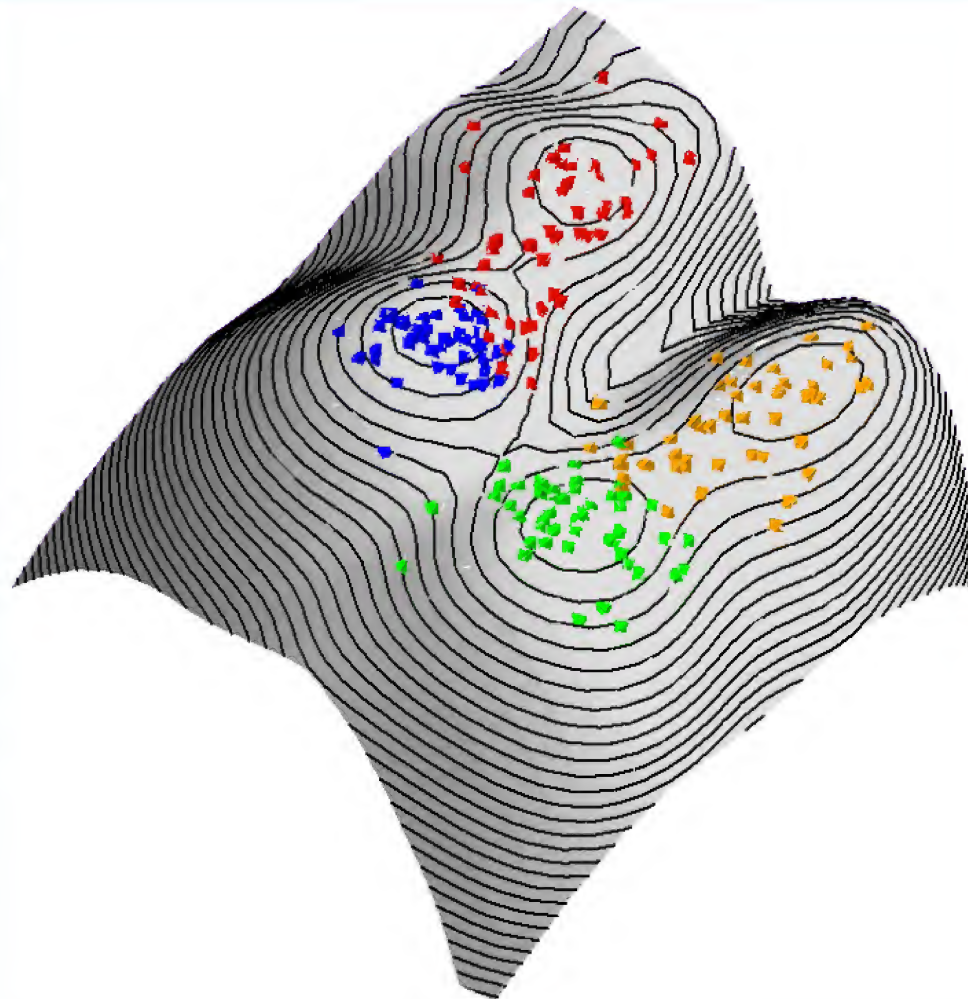
All of Miyamoto-san's design wisdom and experience added to the knowledge base, along with his bizarre princess fetish

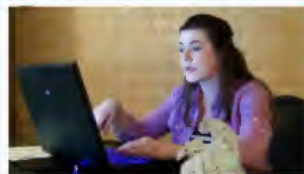
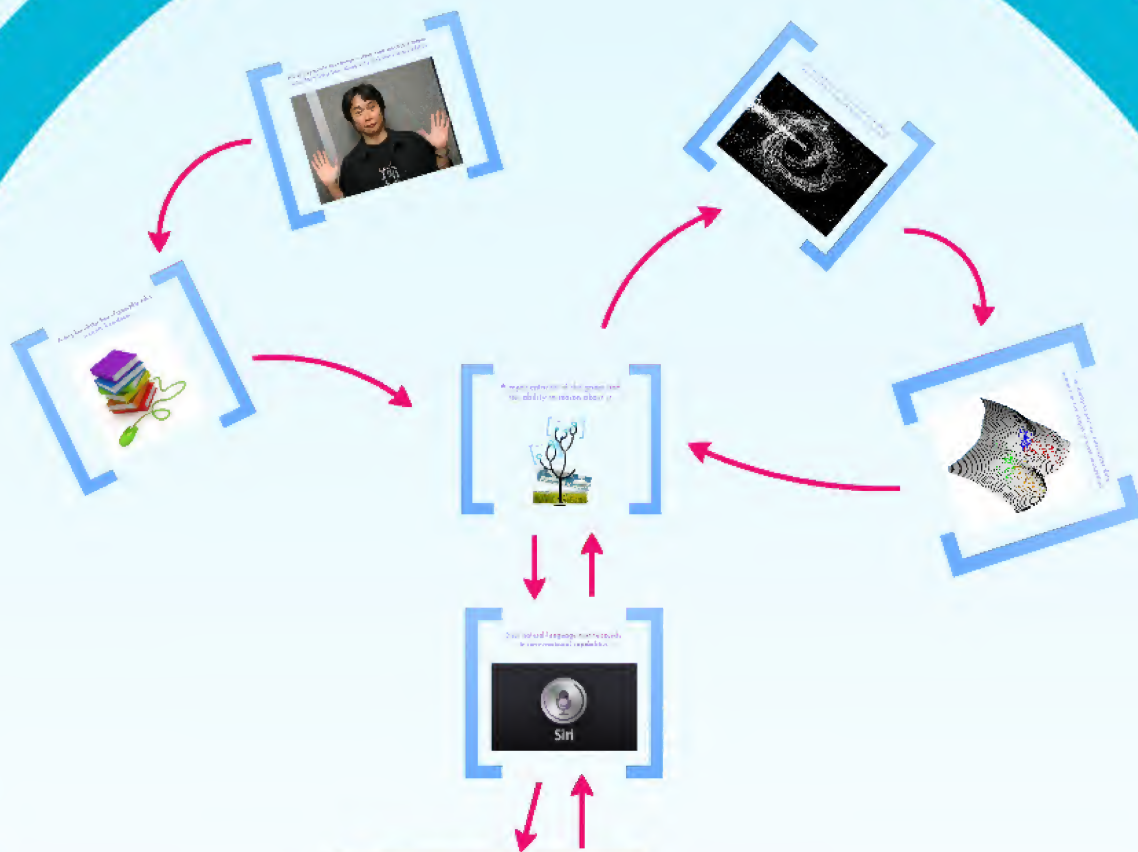
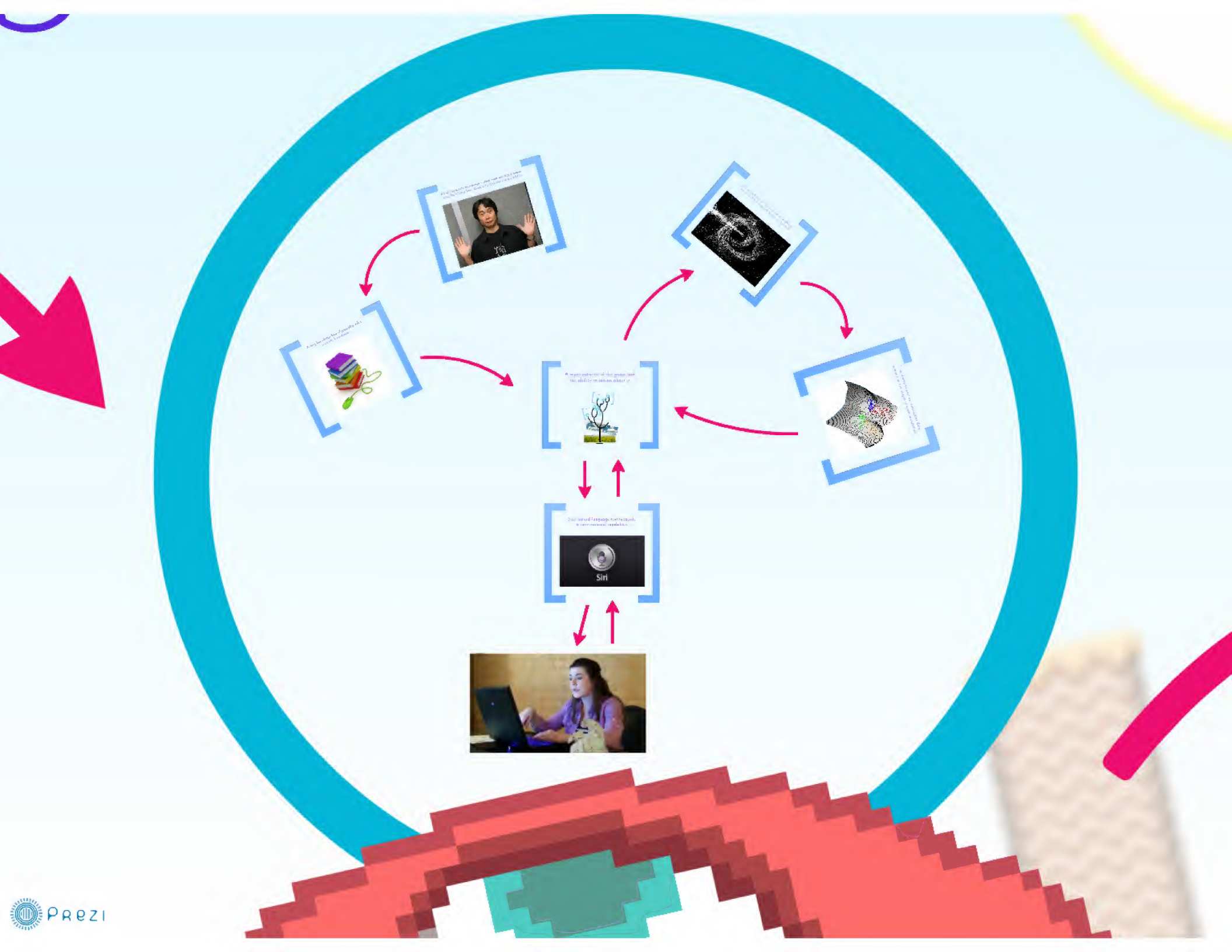


The ability to run millions of simulations of different variations of the design and characterize the results

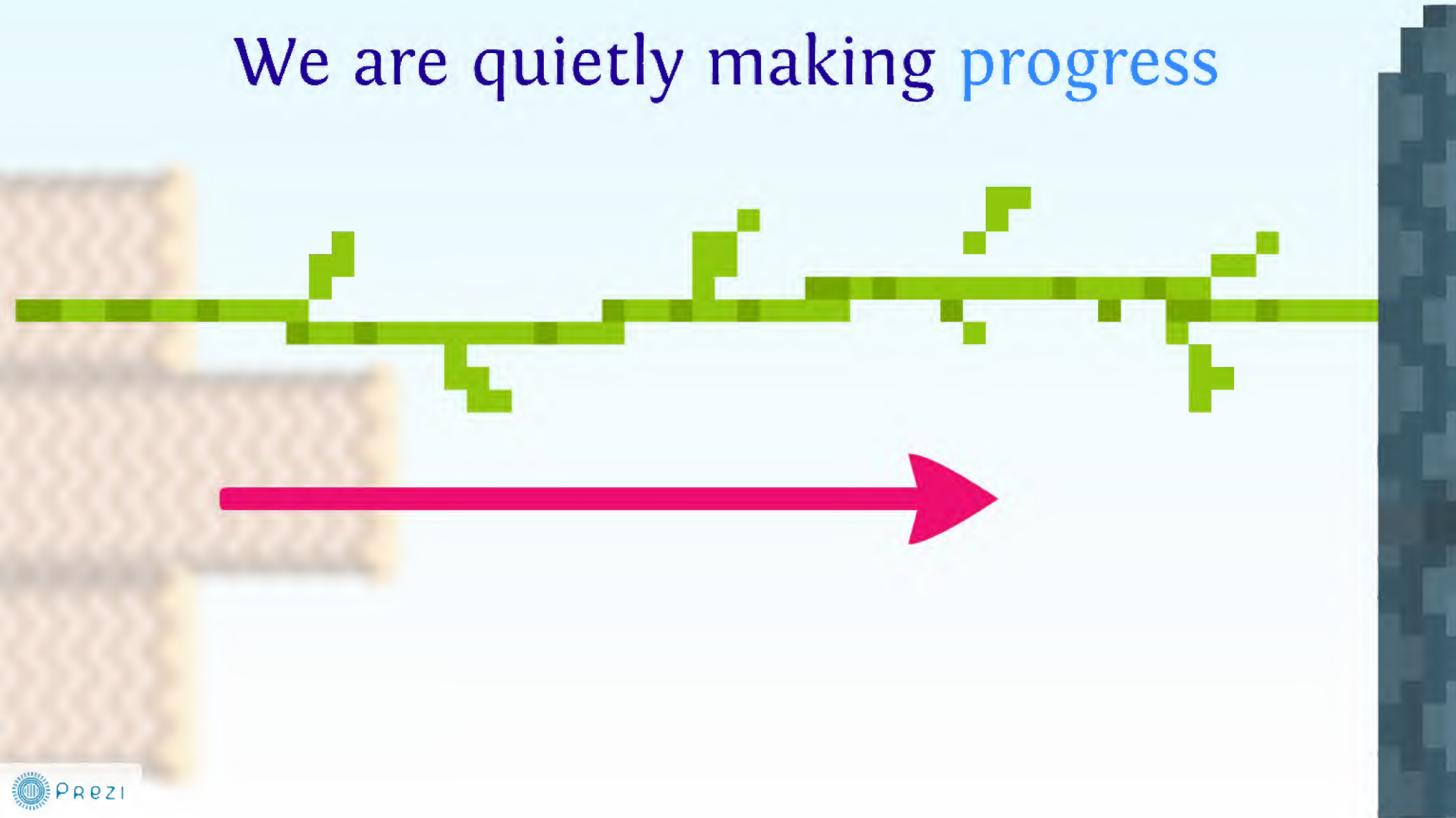


The ability to perform interactive data mining on the output of those simulations





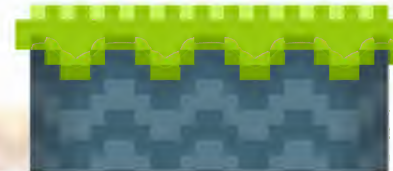
We are quietly making progress



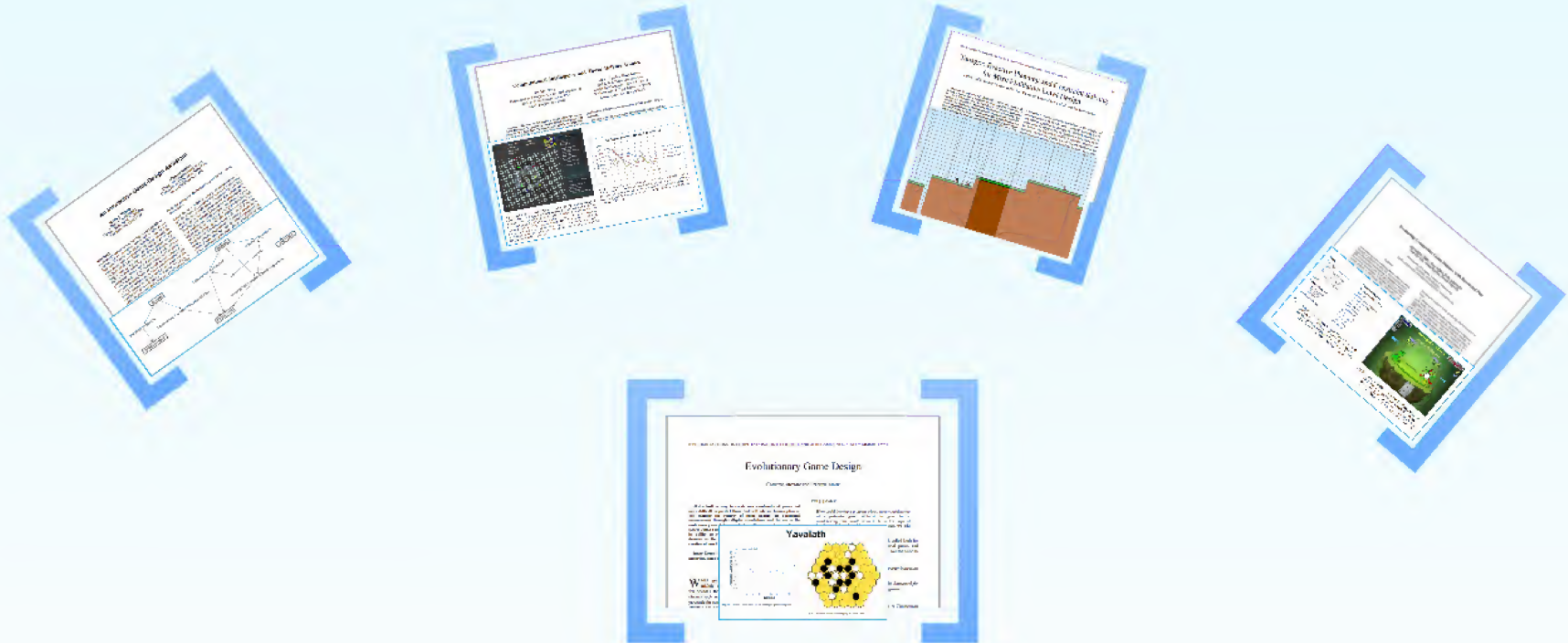
Academic Research



We are quietly making progress



Academic Research



An Interactive Game-Design Assistant

Mark J. Nelson
College of Computing
Georgia Institute of Technology
mnelson@cc.gatech.edu

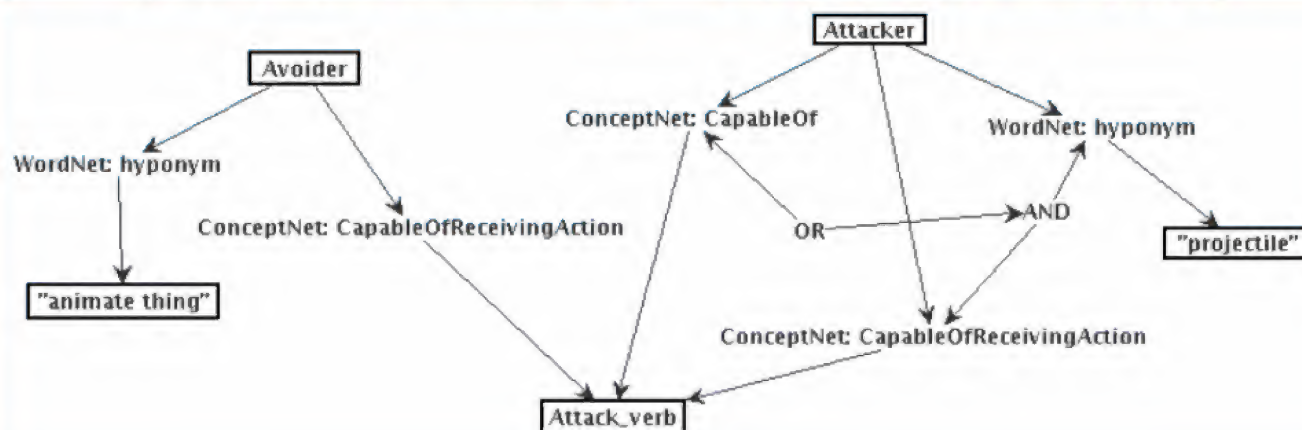
Michael Mateas
Computer Science Department
University of California, Santa Cruz
michaelm@soe.ucsc.edu

ABSTRACT

Game-design novices increasingly hope to use game-like expression as a way to express content such as opinions and educational material. Existing game-design toolkits such as *Game Maker* ease the programming burden, bringing the design of small games within the technical reach of low-budget, non-expert groups. The design process itself remains a roadblock, however: It is not at all obvious how to present topics such as political viewpoints or bike safety in a way that makes use of the unique qualities of the interactive game medium. There are no tools to assist in that aspect of the game design process, and as a result virtually all expressive games come from a small number of game studios founded by experts in designing such games. We propose a game-design assistant that acts in a mixed-initiative fashion,

Flash, that greatly ease the programming burden for creating simple games.

Rather, the main roadblock is the game design process itself: the process of presenting desired content, such as bike safety, the policy implications of a proposed law, or high school algebra, in a way that makes use of the unique qualities of the game medium. These unique qualities include creating a strong sense of player agency, procedurally expressing the possible-worlds implications of player decisions, and enabling players to experientially explore rule systems.² Existing game toolkits such as *Game Maker* or *Alice*³ provide no help with the design problem of mapping content into a system of game rules, but rather facilitate the programming task of implementing the game rules once design is already



Computational Intelligence and Tower Defence Games

Phillipa Avery
Department of Computer Science and Engineering
University of Nevada, Reno, USA
Email: pippa@cse.unr.edu

Julian Togelius, Elvis Alistar,
and Robert Pieter van Leeuwen
Center for Computer Games Research
IT University of Copenhagen, Denmark
Email: {juto, elal, rpvl}@itu.dk

Abstract—The aim of this paper is to introduce the use of Tower Defence (TD) games in Computational Intelligence (CI) research. We show how TD games can provide an important

possibilities to improve the gameplay of TD games using CI methods.

TD games can be real-time or turn-based in nature, and the

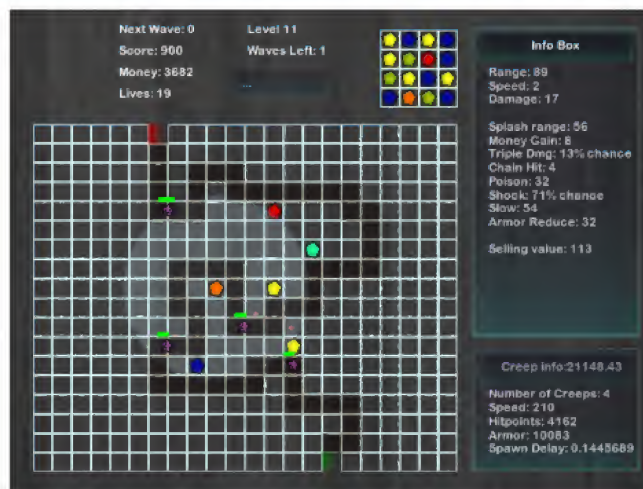


Fig. 4. The Infinite TD game during play of another level, in the middle of a creep wave. The purple stars inside the dark path are creeps, moving from the dark green path entrance towards the red path exit. Above each critter is a small green bar signifying its current health. The brightly coloured pentagons outside of the path are towers. One of the towers is currently selected, meaning that a dimly illuminated circle around the tower shows its range, and the info box to the right of the screen displays its statistics (range, speed, splash range etc).

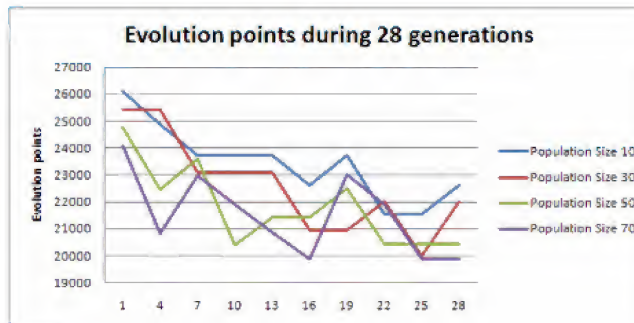


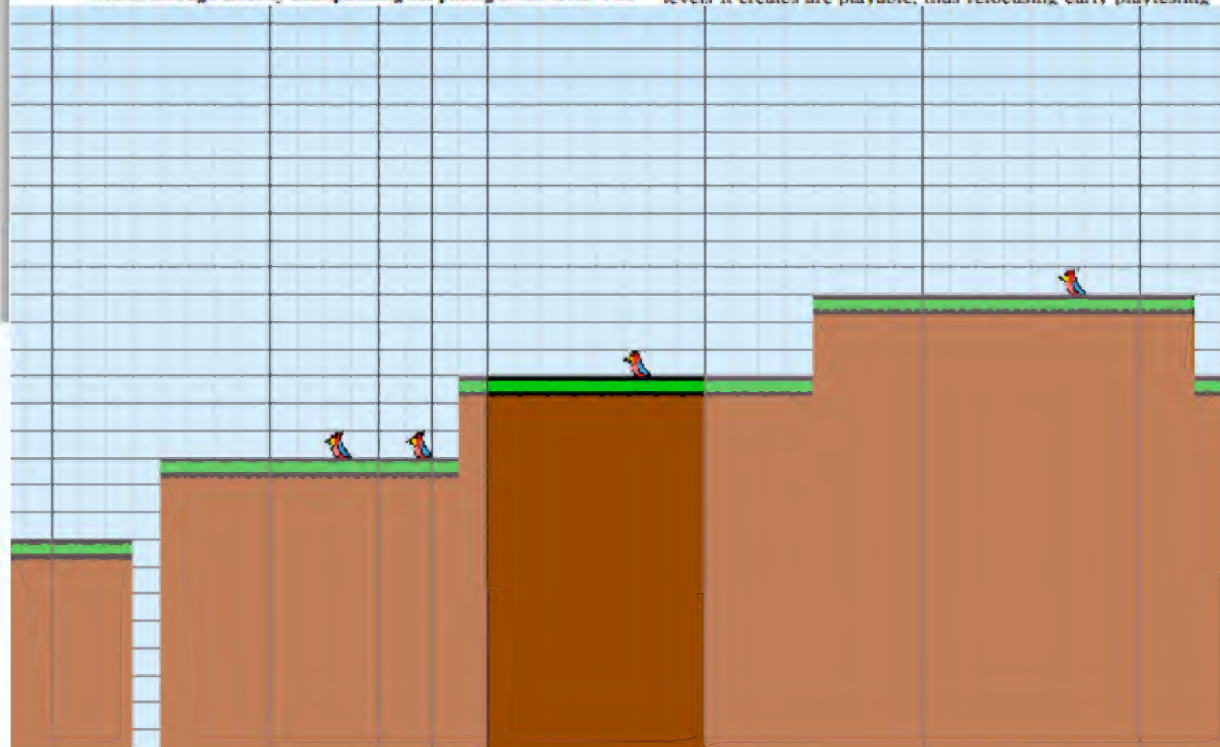
Fig. 5. Creep evolution. The “evolution points” value refers to the number of points that need to be spent before attaining a fitness of at least 1, meaning that at least one creep breaks through the player’s defences and reaches the end of the path. In this game, the creeps are tested against a complicated map with 30 towers.

Tanagra: Reactive Planning and Constraint Solving for Mixed-Initiative Level Design

Gillian Smith, *Student Member, IEEE*, Jim Whitehead, *Senior Member, IEEE*, and Michael Mateas

Abstract—Tanagra is a mixed-initiative tool for level design, allowing a human and a computer to work together to produce a level for a 2-D platformer. An underlying, reactive level generator ensures that all levels created in the environment are playable, and provides the ability for a human designer to rapidly view many different levels that meet their specifications. The human designer can iteratively refine the level by placing and moving level geometry, as well as through directly manipulating the pacing of the level. This

a procedural content generator, capitalizes on the strengths of both human and computer designers. Tanagra's underlying level generator is capable of producing many different variations on a level more rapidly than human designers, whose strengths instead lie in creativity and the ability to judge the quality of the generated content. The generator is able to guarantee that all the levels it creates are playable, thus refocusing early playtesting



Evaluating Competitive Game Balance with Restricted Play

Alexander Jaffe, Alex Miller, Erik Andersen
Yun-En Liu, Anna Karlin, Zoran Popović

Department of Computer Science & Engineering
University of Washington
{ajaffe, amiller, eland, yunliu, karlin, zoran}@cs.washington.edu

Abstract

Game balancing is the fine-tuning phase in which a functioning game is adjusted to be deep, fair, and interesting. Balancing is difficult and time-consuming, as designers must repeatedly tweak parameters, and run lengthy playtests to evaluate the effects of these changes. If designers could receive immediate feedback on their designs, they could explore a vast

and designer intuition, while producing inherently fuzzy answers.

A natural question is whether some portion of competitive game balance evaluation could be automated, through AI simulation by a backend reasoning tool (Nelson and Mateas 2009) rather than playtesting. One major roadblock to this goal is that the standard approach to evaluating balance relies on observation of typical human play, not AI play is not

Cards

Analyze Game ☐ Include Reports

1/1 R

1/2 G

1/6 B

Computed Games

- R1/1, G1/2, B1/6, *2/1.
- R2/5, G3/5, B4/5, +1/3.
- R2/5, G3/5, B4/5.
- R1/1, G1/2, B1/6.

Overview Visualization

Balance Measures

Current Game: R1/1, G1/2, B1/6.

Player 1 (50%)
Aggressive (43% wins, 0.06% fees.)
Low-Randomness (30.72%)
Greedy (22.41%)
Random (22.41%)
Omit R1/1 (11.54%)
Omit B1/6 (11.54%)
Omit G1/2 (11.54%)
No-Randomness (0%)
Prefer R1/1 (0%)
Prefer B1/6 (0%)
Prefer G1/2 (0%)

Fig. 1: A basic analysis using our tool. The results show the performance of varieties of restricted play against a restriction-exploiting player. Players who never play Red 1/1, for example, win 11.54% of the time.



Fig. 2: *Monsters Divided*. G1/1 and B5/6 are played, with +3/2 and +1/3 power cards applied to them, respectively. G's rule is in play: closest to 1/2 wins. $5/6 + 1/3$ beats $1/1 + 3/2$, so P2 wins a B trophy.

Evolutionary Game Design

Cameron Browne and Frederic Maire

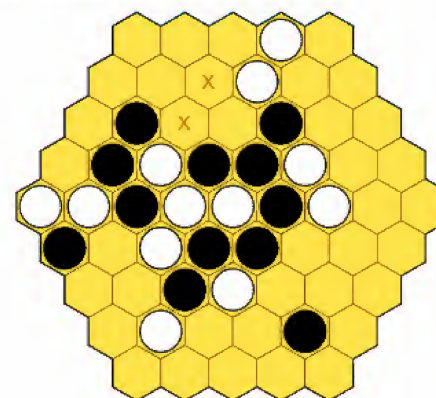
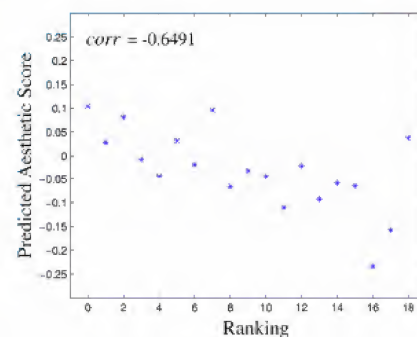
Abstract—It is easy to create new combinatorial games but more difficult to predict those that will interest human players. We examine the concept of game quality, its automated measurement through self-play simulations, and its use in the evolutionary search system called Ludus. We demonstrate the creation of novel

Index Terms—aesthetics, game d

WHILE gar-
artificial i
few decades, the
classics such as
yardstick for suc
attention has bee

Pell [1] states:

If we could develop a program which, upon consideration of a particular game, declared the game to be uninteresting, this would seem to be a true sign of





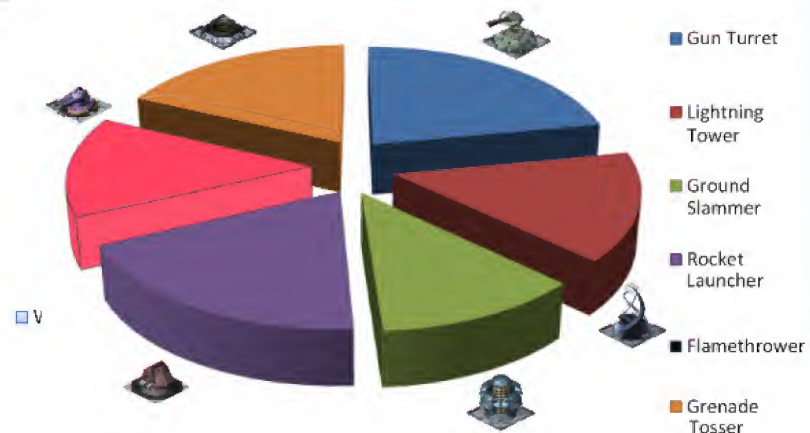
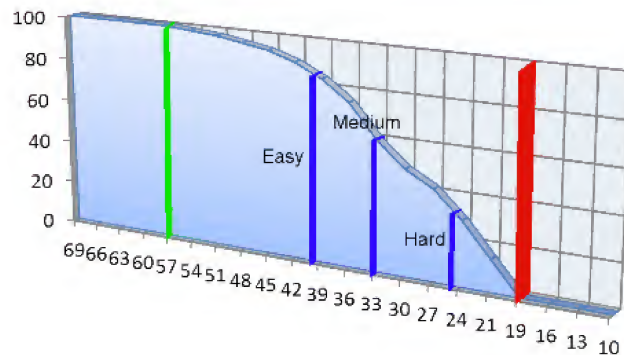
Shigi

Tower Defense Map Balancing

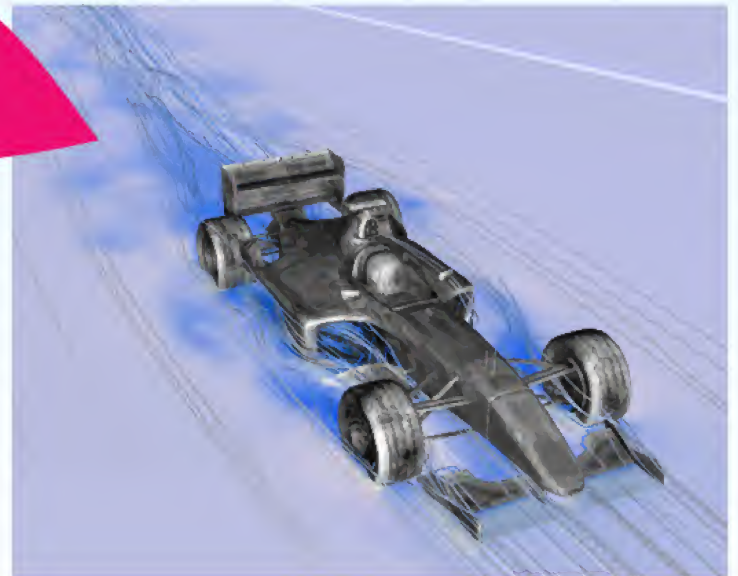


STRONGPOINT TOWER DEFENSE

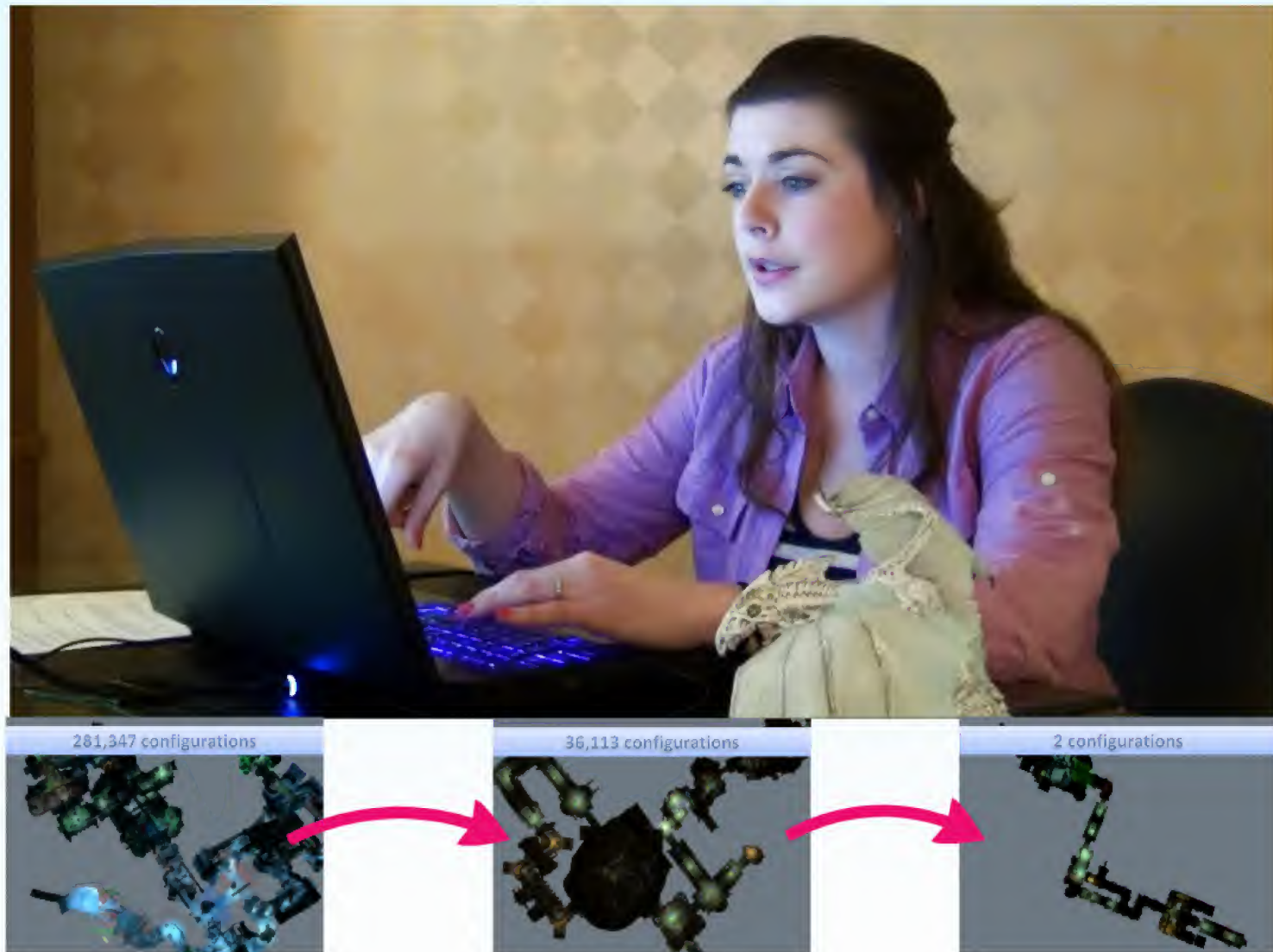
MAP 18

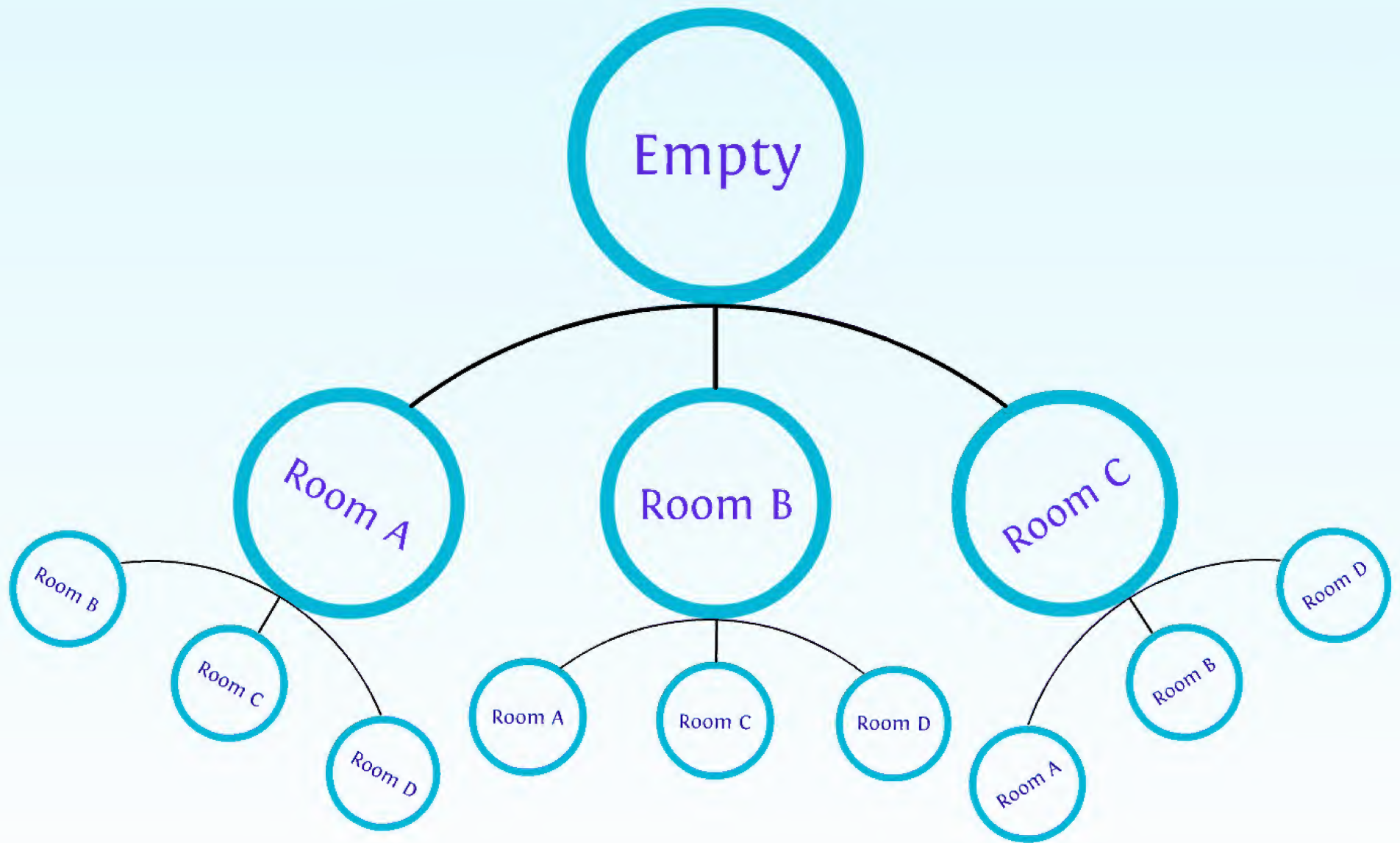


Using Simulation to Determine Airflow of Virtual Wind Tunnel

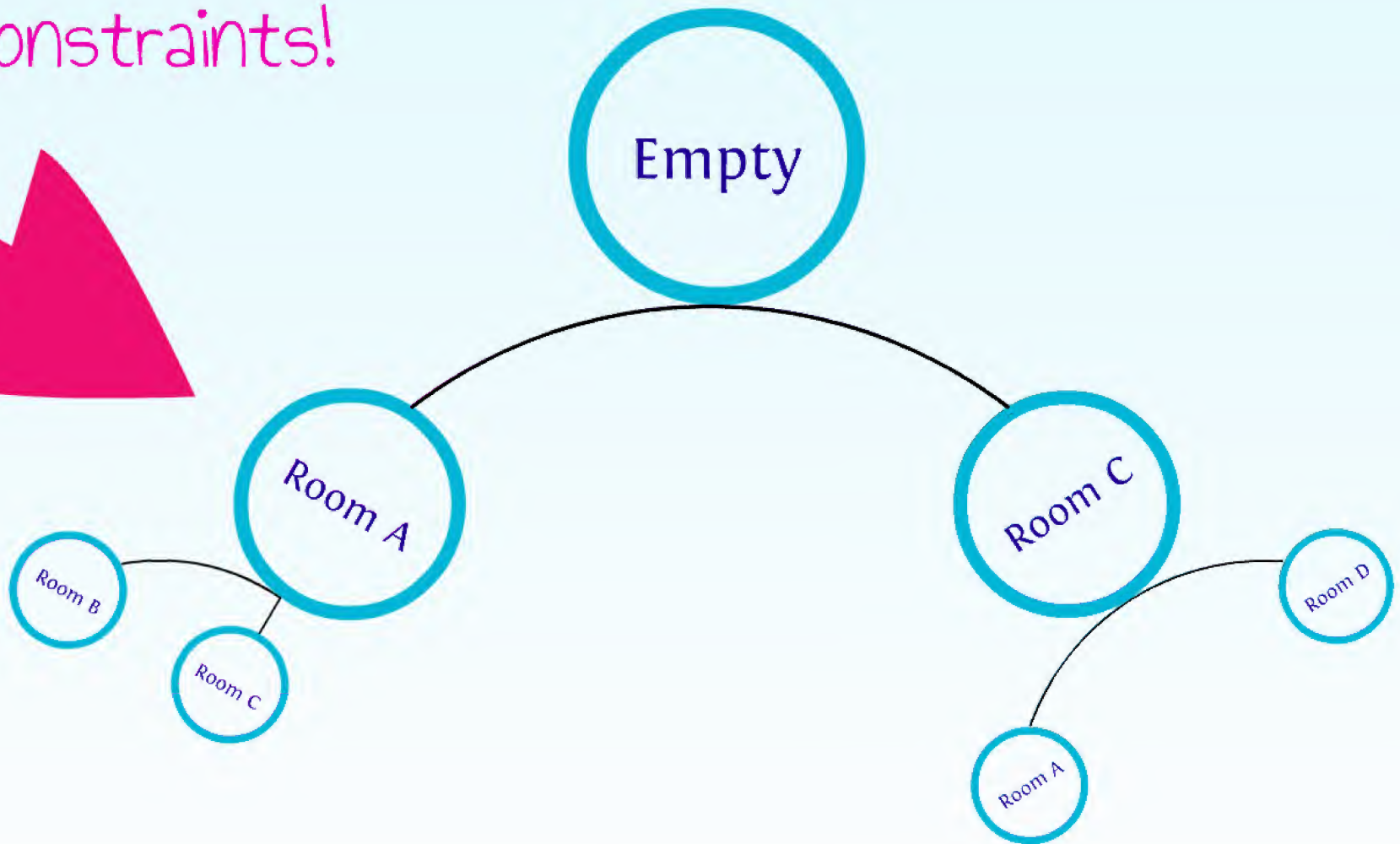


AI-Assisted Reconfiguration of a Game World



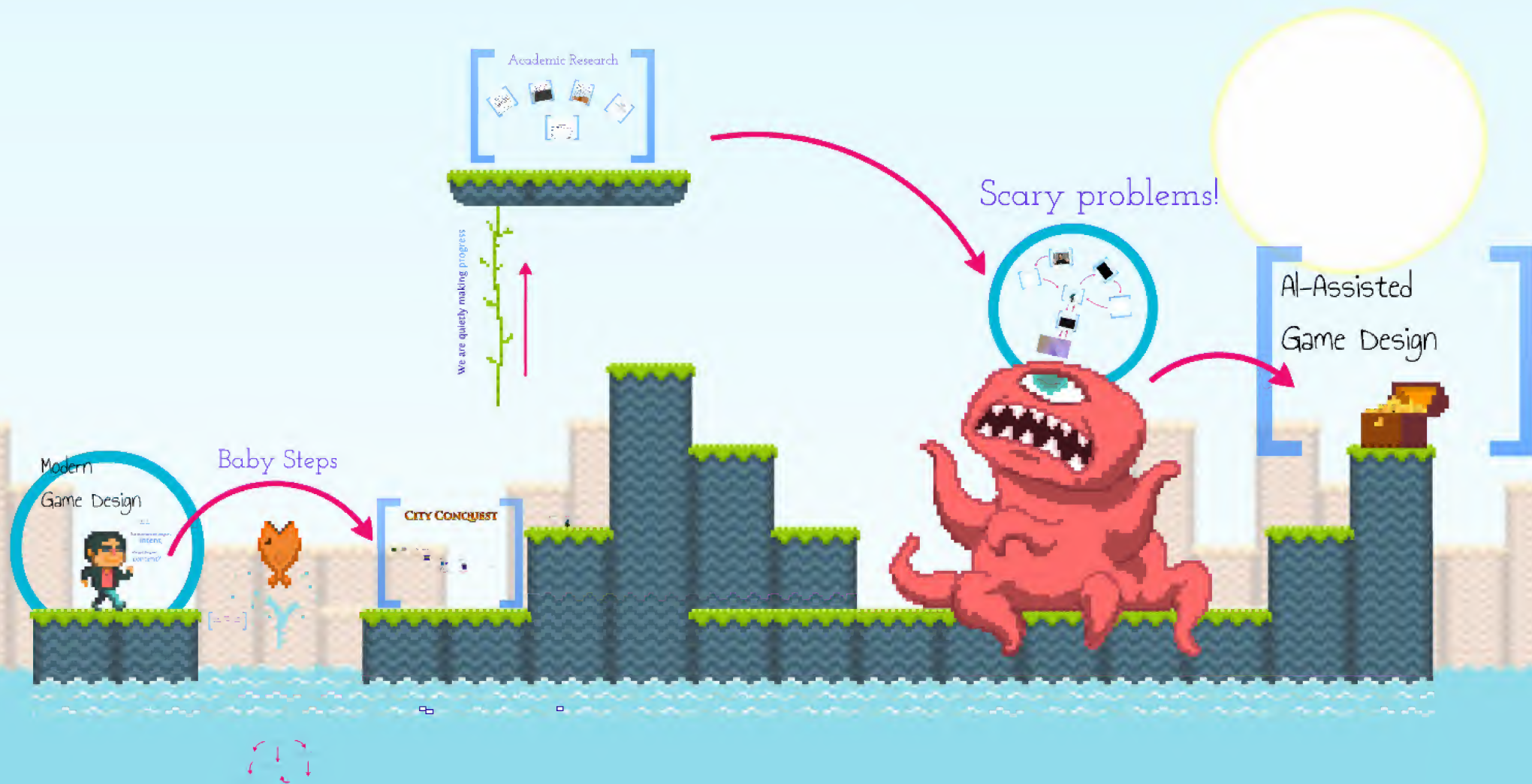


+constraints!



I'm trying to sell you on a way of thinking about design.

I'm not trying to sell you a tool.



Modern Game Design

Baby Steps

Modern Game Design



That understand the designer's intent,
Not just the game's content?

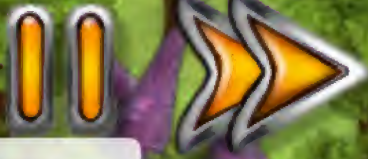


[...]





3





CITY CONQUEST

Design goals

- No globally dominant strategies
- Every tower/unit has a unique tactical role
- Every tower/unit's resource cost (gold or crystals) is proportional to its average utility
- Every tower/unit is useful when considering its unique role and its cost

(... + many more not listed here)



Build=0,11,46,Mine,8
Build=0,11,48,Mine,16
Build=0,11,47,Skyc,24
Build=0,14,42,Turr,28
Build=0,14,40,Turr,171184
Build=0,9,47,Mine,52
Build=0,9,49,Mine,60
Build=0,15,45,GrSl,100808
Build=0,10,49,Drop,100812
Build=0,13,45,RktL,100816
Build=0,17,44,RktL,100820
Build=0,10,45,Skyc,100824
Upgrade=0,13,45,RktL,100816
Build=0,17,49,Drop,100832

...

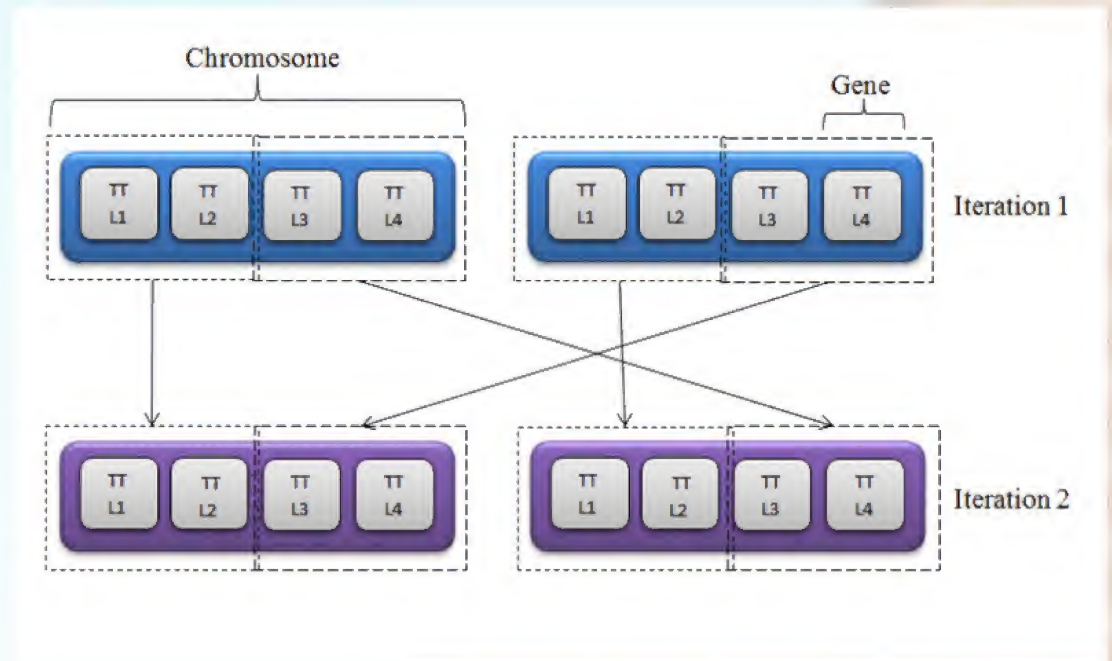
Build=0,11,46,Mine,8
Build=0,11,48,Mine,16
Build=0,11,47,Skyc,24
Build=0,14,42,Turr,28
Build=0,14,40,Turr,171184
Build=0,9,47,Mine,52
Build=0,9,49,Mine,60
Build=0,15,45,GrSl,100808
Build=0,10,49,Drop,100812
Build=0,13,45,RktL,100816
Build=0,17,44,RktL,100820
Build=0,10,45,Skyc,100824
Upgrade=0,13,45,RktL,100816
Build=0,17,49,Drop,100832
...

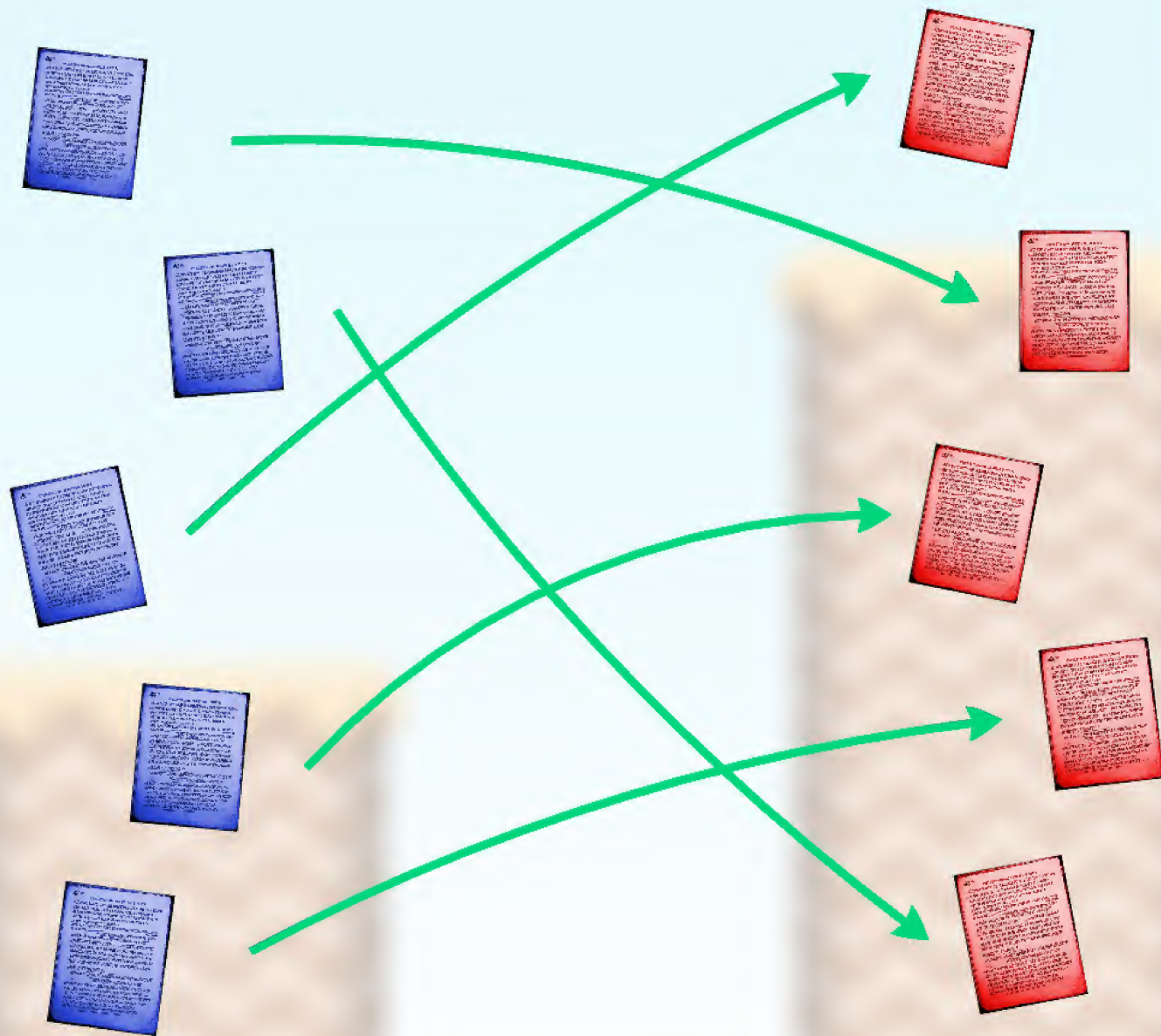
Build=1,11,46,Mine,8
Build=1,11,48,Mine,16
Build=1,11,47,Skyc,24
Build=1,14,42,Turr,28
Build=1,15,45,GrSl,100808
Build=1,14,40,Turr,171184
Build=1,9,47,Mine,52
Build=1,9,49,Mine,60
Build=1,12,44,RktL,100836
Build=1,10,49,Drop,100812
Build=1,13,45,RktL,100816
Build=1,17,44,RktL,100820
Build=1,10,45,Skyc,100824
Upgrade=1,17,44,RktL,100820
...

Mutation

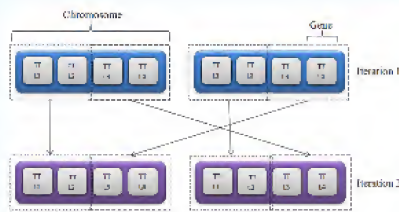
Crossover

Replacement





Mutation
Crossover
Replacement



- Manual re-balancing
- Excel Solver optimization

Aesthetic aspects of balancing are difficult/impossible to quantify

Fitness Function

Player A's Capitol's health

Player B's Capitol's health

Fitness Function

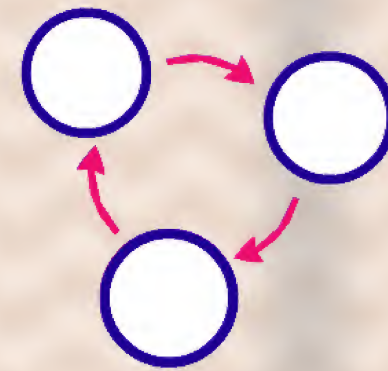
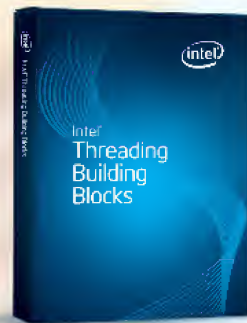
Player A's Capitol's health

Player B's Capitol's health



Evolver

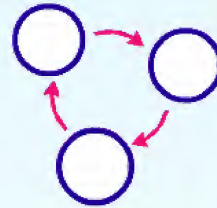
- Skip rendering
- Optimize game logic
- Parallelize with Intel Threading Building Blocks
- Fitness function customizations
- Island genetic model



~1 million simulated games in 12-14 hours
(running overnight on a spare PC)

Evolver

- Skip rendering
- Optimize game logic
- Parallelize with Intel Threading Building Blocks
- Fitness function customizations
- Island genetic model



-1 million simulated games in 12-14 hours
(running overnight on a spare PC)

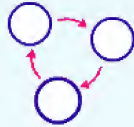
Positive ROI

- Total 2 weeks of coding & tuning effort
- Replaced 3-5 weeks of hand-tuning!
- Better results!
- Allowed incremental exploration of design decisions

Optimized for Victory

Evolver

- Skip rendering
- Optimize game logic
- Parallelize with Intel Threading Building Blocks
- Fitness function customizations
- Island genetic model



-1 million simulated games in 12-14 hours
(running overnight on a spare PC)

Positive ROI

- Total 2 weeks of coding & tuning effort
- Replaced 3-5 weeks of hand-tuning!
- Better results!
- Allowed incremental exploration of design decisions

Optimized for Victory

Computer Player AI

- Traditional AI system
- Behavior tree
- Script-guided building placement hints
- Behavioral mathematics



Optimized for Player
Entertainment (Fun!)

CITY CONQUEST 2

Goals

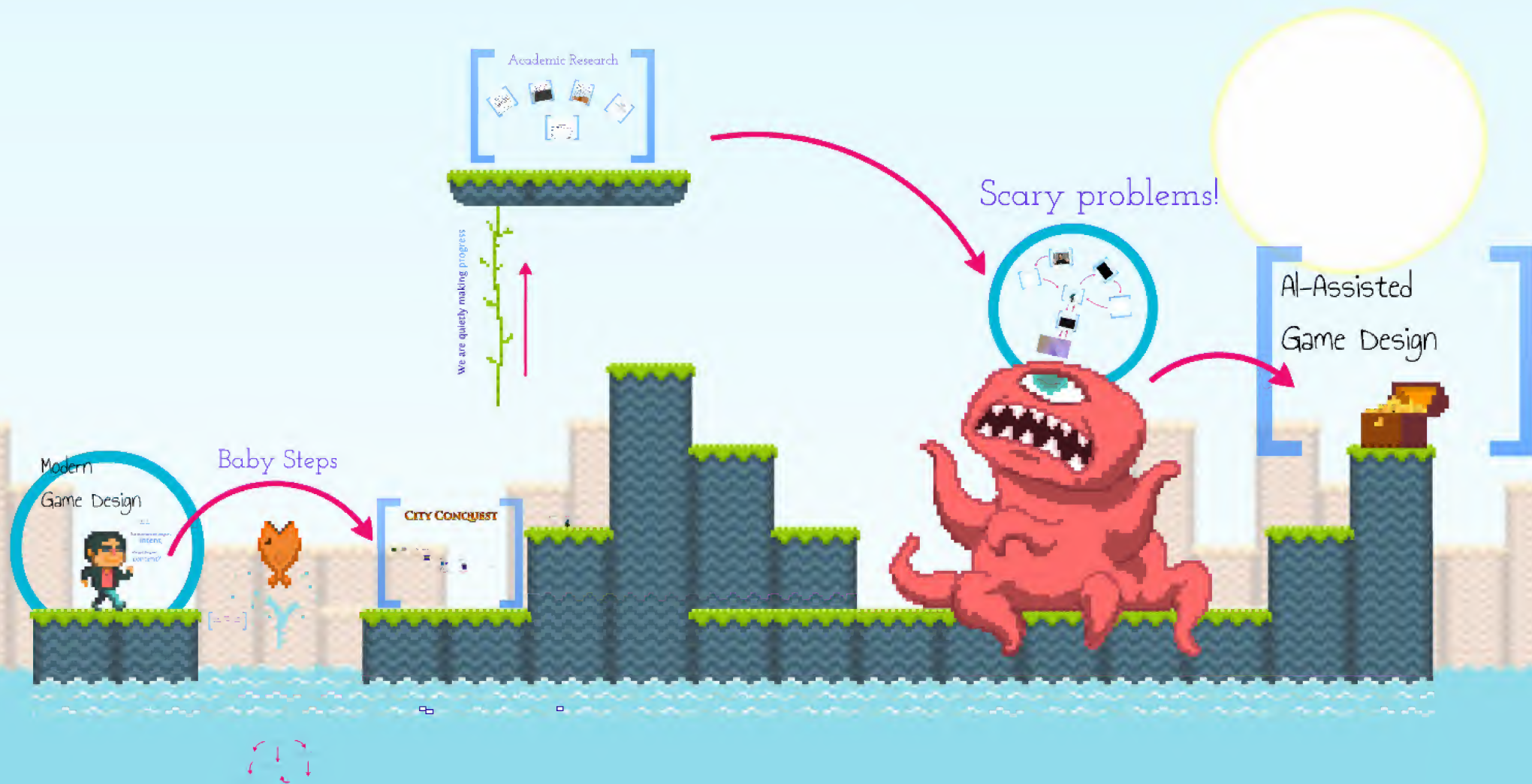
- GA for mission tuning & balancing
- Fitness function customizable for each mission
- GA-based AI vs computer player AI
- Integrate with external data mining system to get more useful data from results

I'm not trying to sell you a tool.

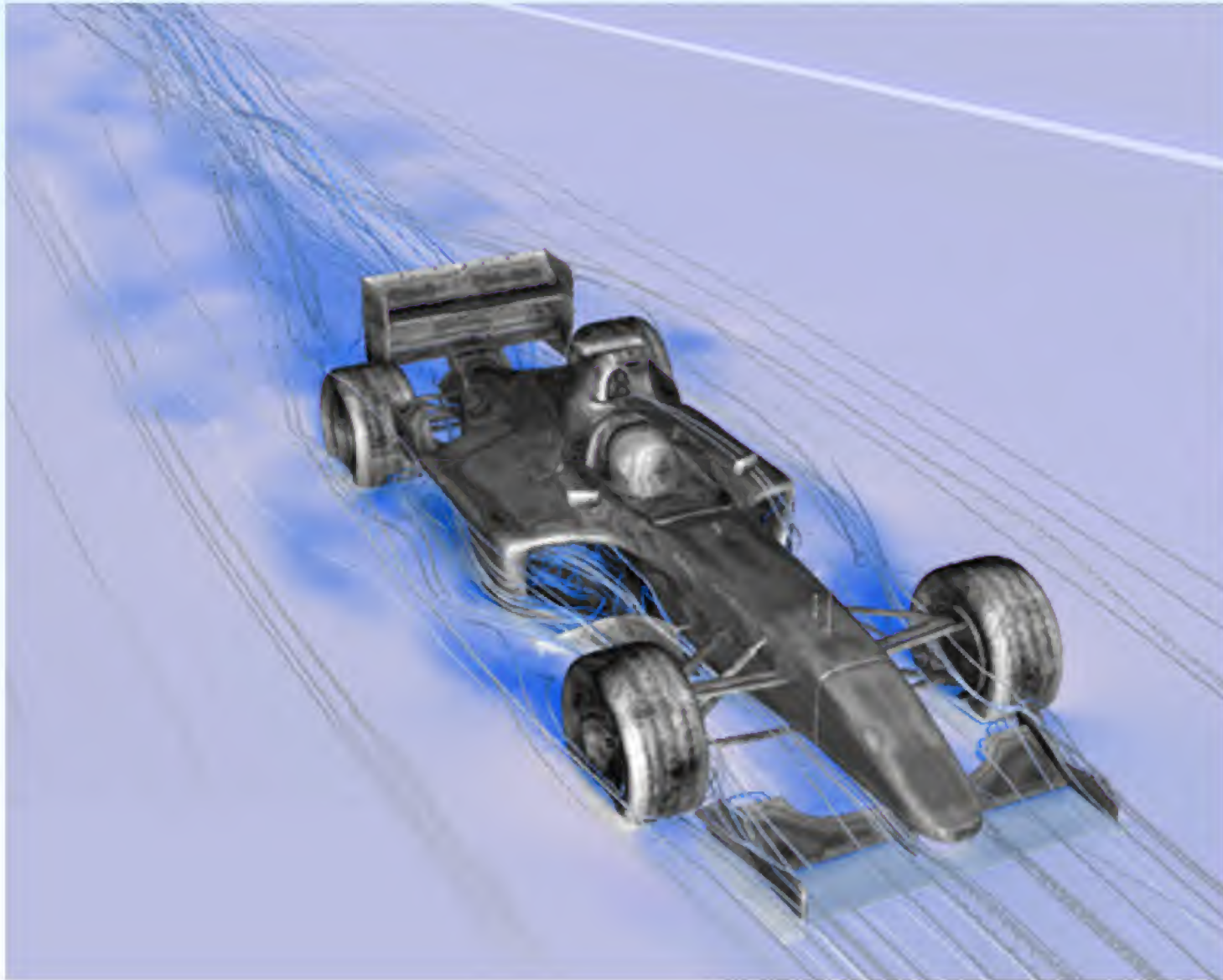
I'm trying to sell you on a
way of thinking about design.

I'm trying to sell you on a way of thinking about design.

I'm not trying to sell you a tool.

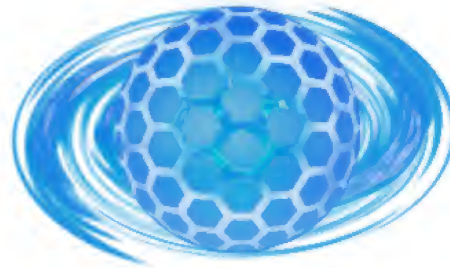


Virtual Wind Tunnel for Game Design



AI-Assisted Game Design





Intelligence Engine
DESIGN SYSTEMS



paultozour.prosite.com
GDC Play kiosk 50

From the Behavior Up: When the AI Is the Design

Paul Tozour

Intelligence Engine Design Systems

Damian Isla

Christian Baekkelund

Moonshot Games

